

Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965
PL ISSN 0860 – 1674

Bajtek

MIESIĘCZNY DODATEK DO "SZTANDARU MŁODYCH" NR 11 LISTOPAD 1986 CENA 100ZŁ





W GRUDNIU: DODATEK SPECJALNY
7 GIER z planszami

HORYZONT POKOLENIA

Przewidywanie przyszłości zawsze było obarczone dużym ryzykiem. „Historia ma odtąd szansę być mniej przypadkową, mniej widowiskową i zarazem mniej dramatyczną. Coraz rzadziej pojawiać się będą wielcy zdobywcy, wielcy reformatorzy, wielcy mężowie stanu" – tak pisał w 1899 roku we Francji Emil Faguet w książce „Jaki będzie XX wiek?". Pisał tak zaledwie na 15 lat przed I wojną światową, na 18 lat przed Rewolucją Październikową, na 40 lat przed najstraszliwszą z wojen... A jednak, mimo zniechęcających przykładów, przyszłością zajmować się musimy. Zaniedbanie myślenia o przyszłości wiąże się bowiem wszędzie z konkretną, wymierną ceną strat społecznych i ekonomicznych. Cenę tę w swoim czasie dobrze zresztą w Polsce poznaliśmy.

Żeby wygrać bezwzględną we współczesnym świecie walkę o rozwój przetrwanie, trzeba – jak w szachach – przewidywać na wiele ruchów naprzód. Jaki jednak minimalny horyzont czasowy myślenia o przyszłości powinniśmy przyjąć?

W 1936 roku w wydawnictwie „Trzaski, Everta i Michalskiego" ukazała się w polskim tłumaczeniu pionierska w zakresie naukowego rozpoznania przyszłości książka Ritchie Galdera „Narodziny przyszłości w retorcie uczonych". Przyszły lord i laureat nagrody im. Kalingi (które to zaszczyty otrzymał za zasługi w popularyzowaniu wiedzy) zaproponował wówczas aby: „określić przyszłość jako świat, z którym będzie miało do czynienia moje pokolenie, biorąc z grubsza – najbliższe trzydziestolecie. Co też uczyni ze światem lub dla niego moja generacja – w ciągu tego okresu?"

I odpowiadał natychmiast: „Wierzę, iż umożliwi wiedzy wykonanie jej zadania – stworzenia naukowej cywilizacji".

Nie stworzyła.

Wystarczy powiedzieć, że z dwudziestu największych wynalazków okresu powojennego zaledwie jeden, to znaczy magnetowid, został wymuszony przez potrzeby rynku cywilnego. Pozostałe dziewiętnaście wynalazków ma pośrednie lub bezpośrednie zastosowanie wojskowe. Ale to na marginesie.

Przypomniałem tutaj książkę Caldera, aby przyłączyć się do zaproponowanego przez niego „horyzontu pokolenia". Nie jest to oczywiście horyzont jedyny z możliwych. Ale jego zaletą jest pogodzenie dwóch spojrzeń na przyszłość – społecznego oraz indywidualnego. Rola czynnika pokoleniowego będzie zresztą z roku na rok rosła. Ponad 36 proc. ludności świata nie ukończyło jeszcze 15 lat, a 60 proc. nie ma jeszcze lat 30 – warto o tym pamiętać.

Z tak określonego horyzontu widzenia przyszłości wynika wiele wniosków praktycznych. I to wniosków na dzisiaj:

Po pierwsze – w naszym myśleniu o przyszłości należy przekroczyć wreszcie zaczarowaną barierę roku 2000. Bo co w końcu takiego zdarzy się w noc sylwestrową pomiędzy rokiem 2000 a 2001 (wtedy bowiem dopiero rozpocznie się wiek XXI)? Zmieni się rok, zmieni się wiek, zmieni się tysiąclecie. I jakich magicznych interpretacji by do tego nie dorabiać, to żadnych nieciągłości rozwoju na pewno wówczas nie będzie. W roku 1000 też zresztą nie było końca świata, mimo że dosyć powszechnie taką możliwość przewidywano!

Po drugie – aby nie płacić ponownie frycowego – myślenie kategoriami przyszłości musi być stale obecne w naszym życiu społecznym. Nie „jakoś to będzie", nie „aby do jutro"... Widzenie przyszłości w całym jej pokoleniowym horyzoncie jest na dzisiaj absolutną koniecznością!

Po trzecie – warto zdać sobie sprawę, i dotyczy to już bezpośrednio czytelników „Bajtka", że nie istnieje nic takiego, jak jakaś „obiektywna" przyszłość, która kiedyś przyjdzie, wystarczy tylko usiąść i poczekać na nią. Przyszłość będzie taką, jaką ją swoimi kolejnymi, dzisiejszymi działaniami stworzymy. Wniosek ten pociąga za sobą istotne implikacje. Bo skoro wiemy, że nie uda nam się na nikogo zrzucić odpowiedzialności za naszą przyszłość, to powinniśmy dzisiaj zrobić wszystko, aby stworzyć sobie jak najszersze pole działania za lat 5, 10, 20... I nie mieć potem do nikogo pretensji, gdyby to pole okazało się za małe.

Po czwarte – i to jest wniosek nie tylko dla nastolatków – najopłacalniejszą przyszłościowo formą inwestycji jest inwestowanie w ludzi. Bo konkretne techniki i technologie mogą się zmieniać, mogą pojawić się (i na pewno pojawią) nowe dziedziny wiedzy i produkcji. Ale wszechstronnie wykształcony człowiek nie przeżyje się nigdy. Dlatego popierać będziemy gorąco w „Bajtku" te wszystkie formy działań, które służą popieraniu talentów, pomaganiu ludziom odważnym, nowatorskim, potrafiącym myśleć i działać niekonwencjonalnie. Taki jest główny cel istnienia Młodzieżowej Akademii Umiejętności powołanej z inicjatywy „Sztandaru Młodych". Taka jest również zasadnicza myśl przewodnia przeboju ostatnich wakacji, czyli zorganizowanych z inicjatywy Biura Urzędu Rady Ministrów ds. Młodzieży cyklu obozów pod hasłem „Awangarda XXI wieku". Dla tych, którzy wzięli w tych obozach udział, było to unikalna możliwość przedyskutowania wszystkich spraw które ich interesują. Spraw, które ważne są już dziś, a których waga w horyzoncie życia pokolenia będzie stale rosła. W przyszłym roku „Bajtek" postara się zresztą wziąć patronat na jednym z takich obozów!

Po piąte wreszcie – aby wygrać przyszłość niezbędne jest właściwe rozpoznanie dominujących w niej trendów. W wielu sprawach możemy się pomylić, ale na pewno nie grozi nam ryzyko błędu w sprawie informatyki. Od tej sprawy odwrotu nie będzie, choć zapewne komputery naszych dzieci w niczym nie będą przypominać naszych wspaniałych maszyn typu „Spectrum", „Amstrad" czy „Commodore". Ale nie martwcie się – w Muzeum Techniki wywalczymy dla tych poczciwców najbardziej honorowe miejsce!

Waldemar Siwiński



WYBIERZ SAM

GRA O JUTRO
Kompromis 3
SWEGO NIE ZNACIE
Komputery na Okęciu 4
Bity zamiast czcionek 5
PROGRAMOWAĆ MOŻE KAŻDY
PASCAL cz.2 6
KLAN AMSTRAD SCHNEIDER
Polskie litery 8
Trochę ruchu 8
Pamięć pod lupą 9
KLAN SPECTRUM
Słupki 10
Renumeracja 10
Kłopoty z pamięcią 10
Master Toolkit 11
Czy jesteś jasnowidzem? 12
KLAN ATARI
Tape copier 14
Świat dźwięków 15
KLAN COMMODORE
Poradnik młodego pirata cz. 4 20
SAM 20
TSL Copy 20
Lister 21
Polski alfabet cz. 2 22
PRZED EKRANEM
Rachunki i tabelki 13
CO JEST GRANE
Enigma Force 16-18
Globtroter 19
Zorro 19
UWAGA PIRACI 24
JAK TO ROBIĄ INNI
Co słychać w Szwecji 25
SAMI O SOBIE
O Złotą Dyskietkę „Bajtka" 28
Radiokomputer 28
HOBBYTE 28
Forum Atari 28
GIEŁDA 30
TYLKO DLA PRZEDSZKOLAKÓW
Kubuś Telegrafista 31
NIE TYLKO KOMPUTERY
EXPO'86 32

„BAJTEK" – MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH".
ADRES: 00–685 Warszawa, ul Wspólna 61. Telefon 21–12–05.
Przewodniczący Rady Redakcyjnej: Jerzy Do-mański-redaktor naczelny „Sztandaru Młodych".
ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego "SM" — kierownik zespołu "Bajtka"), Roman Poznański (z-ca sekretarza redakcji "SM" — sekretarz zespołu „Bajtka"), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Kowalewski, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski.
Klany redagują: Commodore — Klaudiusz Dybowski, Michał Silskl Amstrad-Schneider — Tomasz Pyć, Sergiusz Wolick Spectrum — Konrad Fedyna, Michał Szuniewicz Atari — Wiesław Migut
WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403,414. Cena 100 zł. Skład techniką CRT-200, przygotowalnią offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.
Zam. nr 1301/86, nakład 250 000 egz. P-101

SM SZTANDAR MŁODYCH

Bajtek



KOMPROMIS

Rozmowa z WOJCIECHEM SZANTEREM – dyrektorem Zarządu Centralnej Składnicy Harcerskiej.

— Pięć tysięcy komputerów firmy Timex — TC 2048, trzy i pół tysiąca drążków sterowych „Quickshot", piętnaście tysięcy trzycalowych dyskietek DS/DD, dwa tysiące komputerów SVI-738 produkcji Spectravideo — to tylko część sprzętu sprowadzanego przez Centralną Składnicę Harcerską. Czy jest to pierwsza tego typu transakcja na naszym rynku?

— Nie jest istotne, czy my byliśmy pierwsi. Transakcje związane z rękojmią i prowadzeniem sprzętu zawierają „Baltona" i „Pewex". Rynkowym sprzedawcą jest zielonogórska „Apina" czy Dom Handlowy Nauki PAN. Natomiast jeśli chodzi o sklepy ogólnodostępne — to rzeczywiście, pod względem rozmiarów jest to pierwsza tego typu transakcja.

— Przymiarki, aby sprowadzić sprzęt komputerowy, trwały od dawna. Krążyły różne pogłoski o jakości sprzętu do zakupienia, o jego cenach, ilości. Doszło wreszcie do sfinansowania pierwszej umowy — za co chwała Centralnej Składnicy Harcerskiej. Czy mógłby pan odsłonić kulisy, jak doszło do zawarcia transakcji? Jest przecież wiele firm o lepszej renomie niż Timex czy Spectravideo.

— Mimo pozorów kulisy nie są aż tak ciekawe. Wolałbym jednak o nich nie mówić, gdyż jako handlowca obowiązuje mnie zachowanie poufności — chociażby w transakcjach importowych. Po prostu te firmy udzieliły nam najdogodniejszych warunków. Nam — to znaczy Towarzystwu Handlu Międzynarodowego DAL — centrali, która dokonała zakupów. Ostateczne zawarcie umowy było poprzedzone licznymi sondażami, konsultacjami, badaniem rynków. Producenci krajowi, jako dostawcy sprzętu są dla Składnicy nieosiągalni, chociażby ze względu na oferowane ceny i znikomą wielkość produkcji. Również nie dochodziła do skutku fuzja z pewnymi firmami polonijnymi — za bardzo troszczyły się o własne interesy finansowe. Rozpoczęliśmy więc rozmowy z przedsiębiorstwami „Metronex" i DAL. „Metronex" mógł sprowadzić komputery tylko na zasadzie transakcji kompensacyjnej. Jedynie DAL miało możliwości dokonania zakupów. Po licznych wywoławczych ofertach z naszej strony i po zagranicznych konsultacjach cenowych zdecydowaliśmy się na transakcję właśnie z Timex'em i Spectravideo. Chociaż wśród kontrahentów były także Amstrad czy Commodore. Jednakże ze względu na to, że firmy te sprzedają w Polsce komputery w sklepach „Pewexu" i „Baltony", musieliśmy zrezygnować z ich propozycji. Tak więc, biorąc pod uwagę nasze możliwości zakupu oraz możliwości sprzedaży na rynku krajowym sprowadziliśmy ten właśnie sprzęt.

— A więc sukces?

— Bez przesady, nie jest to szczyt szczęścia i naszych marzeń. Ale trzeba mieć świadomość, że w handlu dokonuje się wyboru i kompromisów: kupuje się to na co nas stać, to co oferują inni lub... nie kupuje się nic. Na przykład długo trwały pertraktacje na temat cen komputerów relacji ceny osprzętu do jednostki podstawowej. THM DAL nie mogło wyasygnować wysokich kwot, które chciałoby się przeznaczyć na zakup bogatego oprogramowania dla MSX czy różnorodnych urządzeń peryferyjnych. Uważam jednak, że mimo tych ograniczeń, sprowadziliśmy sprzęt przyzwoitej jakości, który można będzie kupić po godziwej cenie. Ceny zostały tak skalkulowane, że obniżyliśmy należną nam marżę o 16 proc. do 12 proc. a w niektórych wypadkach nawet poniżej tej granicy. Wartość transakcji w cenach detalicznych ok. 3,4 mld. zł.

— Klienta jednak nie interesuje możliwości dewizowe central handlowych czy układy w polskim rynku komputerowym. Jeśli wydaje kilkaset tysięcy złotych, to chce mieć sprzęt dobrej jakości.

GRA O JUTRO

Tymczasem urządzenia firmy Spectravideo praktycznie nie funkcjonują na naszym rynku. Timex to inna wersja Spectrum, które nie cieszy się już zbyt dużą popularnością.

— Model TC 2048 jest w pełni kompatybilny ze Spectrum plus, natomiast komputery Spectravideo pracują w systemie CP/M — nie powinno być więc większych problemów z wykorzystaniem już istniejącego oprogramowania. Do TC 2048 może być z powodzeniem wykorzystywana cała rodzina gier i programów edukacyjnych.

— Pewne wątpliwości może jednak budzić kupowanie sprzętu, który posługuje się dyskietkami 3,5", których u nas nie ma. Obowiązujący wśród naszych fanów mikroinformatyki standard — 5,25".

— Dlatego do komputera SVI 738, który ma wbudowaną stację dysków 3,5" sprowadzamy także zewnętrzną stację dysków 5,25". Dyskietki 3" sprowadzane są do stacji dysków FDD3" dla komputera TC 2048. Tym samym formatem posługuje się również Amstrad — tak popularny u nas.

Sprzęt był testowany przez kluby komputerowe m.in. Katowicki Informik, badany w instytutach badawczych, a także przez indywidualnych specjalistów. Zaznaczam, że jest to pierwsza, sondażowa partia sprzętu. Jeśli sprzedaż będzie bez kłopotów, mamy od dostawców Spectravideo zapewnienia rozszerzenia asortymentu. Na przykład 200 egzaminatorzy SVI 788 ma klawiaturę angielską. Następna partia będzie wyposażona w klawiaturę polską. Już w transakcji zapewniono sprzedaż 10 programów użytkowych, i ok. 50 edukacyjnych, które będą w pełni dopuszczone do kopiowania. Tak więc w uruchamianej przez nas sieci sklepów oraz hardware'u będzie także software.

— Jak ocenia pan szanse sprzedaży na naszym rynku?

— Początkowo miałem pewne obawy związane z firmą Spectravideo, ale już w połowie września na miesiąc przed planowanym otwarciem stoisk, mieliśmy tyle zamówień, że wszelkie niepokoje okazały się płonne. Na przykład napłynęły zamówienia na wszystkie dyskietki trzycalowe. Różne firmy chcą je wykupić, aby później sprzedawać z nagranymi programami. Oczywiście na taki układ nie możemy sobie pozwolić.

Od początku swego istnienia Centralna Składnica Harcerska miała za jeden z głównych celów rozwijanie zainteresowań politechnicznych młodzieży. Konsekwencją tego założenia jest dokonana transakcja.

— Miejmy nadzieję, że ta transakcja przełamie bariery niemożności związane ze sprowadzaniem sprzętu komputerowego. Traktujemy ją jako szansę rozpoczęcia państwowej sprzedaży — konkurencyjnej wobec giełdy. W najbliższych numerach „Bajtka" bliżej przedstawimy komputery TC 2048 i SVI 738 wraz z osprzętem.

Rozmawiali: Sławomir Polak
Roman Wojciechowski

WYKAZ ASORTYMENTOWY SPRZĘTU MIKROKOMPUTEROWEGO OFEROWANEGO DO SPRZEDAŻY W PLACÓWKACH HANDLOWYCH CENTRALNEJ SKŁADNICY HARCERSKIEJ

Lp.	Wyszczególnienie	Ilość w szt.	Cena w tys. zł
I	**Sprzęt firmy TIMEX**		
1.	Komputer TC 2048	5000	106
2.	Stacja dysków TIMEX FDD 3"	1500	255
3.	Drukarka TIMEX 50	700	96
4.	Drukarka TIMEX 1000	500	280
5.	Magnetofon kasetowy TIMEX 2020	2000	30
6.	Joystick QUICKSHOT	3500	8,5
7.	Dyskietka 3" DS/DD	15000	3
II	**Sprzęt firmy SPECTRA VIDEO (MSX-2)**		
1.	Komputer SVI — 738	2000	440
2.	Stacja dysków BW112 5,25"	400	220
3.	Monitor mono TTL12"	1700	94
4.	Monitor kolor DCM-414,13"	300	262
5.	Drukarka Centronics GLP z kab.	1000	300
6.	Magnetofon kasetowy 767	200	33
7.	Plotter Sony z kab.	300	415
8.	Karta pamięci zewn. 64 KB	100	57
9.	Zestaw graf. EDDY II + CAT	400	90
10.	Mikrodyski 3,5" (po 10 szt.)	4000	36
11.	Kaseta z taśmą do drukarki GLP	2000	8

SWEGO NIE ZNACIE



KOMPUTERY — NA — OKĘCIU

Jeżeli można mówić o komputeryzacji jako o czekającej każda dziedzinę życia czy gospodarki konieczności, to w lotnictwie jest to konieczność szczególna. Warszawskie lotnisko międzynarodowe nie zostaje w tyle za światowa czołówka — na komputery można sie natknąć dosłownie na każdym kroku, w kasach biletowych, biurze reklamacji bagażu itd.

Dzisiaj zajmiemy się jednak komputerami, których zwykły pasażer nie widzi, lecz bez których nie byłoby mowy o nowoczesnym lotnictwie, czy nawet o jego namiastce.

KONTROLER CZUWA

Jesteśmy w długiej, zaciemnionej hali zastawionej pulpitami, na których aż się roi od przycisków, telefonów, migających kontrolek itp. Pomiędzy pulpitami widać okrągłe ekrany radarowe. Siedzą przy nich ludzie ze słuchawkami na uszach — to kontrolerzy ruchu lotniczego, od których zależy bezpieczny lot wszystkich samolotów komunikacyjnych. Pod ich palcami jarzą się podświetlane klawiatury połączone z komputerem firmy **PDP**.

Trudno nawet wyobrazić sobie jak wielka lawina informacji przewala się codziennie przez **Centrum Kontroli Ruchu Lotniczego** — opisana powyżej hala to właśnie jego część; mieści się tu kontrola zbliżania i kontrola obszaru.

Zainstalowany na lotnisku Okęcie system radiolokacyjny jest wyposażony w dwa radary: pierwotny i wtórny. Radar pierwotny pozwala na określenie pozycji każdego samolotu znajdującego się w powietrzu, zaś radar wtórny pozwala przy takim echu radiolokacyjnym „doczepić" kilka istotnych i bardzo ważnych dla kontrolera informacji; najważniejsze jest to, że informacje te dotyczą czasu realnego.

W każdym samolocie komunikacyjnym zainstalowane jest urządzenie zwane transponderem, odpowiadające na sygnały wysłane z anteny radaru wtórnego. Dzięki temu obok echa pierwotnego pojawia się także echo wtórne z pakietem informacji dotyczących aktualnej wysokości,prędkości śledzonego samolotu i jego numeru identyfikacyjnego, (**squawk**). Kontroler ma także możliwość przypisania dla każdego z samolotów na ekranie radaru numeru rejsu, kategorii pozostawianego śladu aerodynamicznego (zawirowania powietrza za samolotem) czy innych informacji. Te udogodnienia są oczywiście zaledwie ułamkiem możliwości komputera. Może on także kontrolować utrzymywanie się samolotu w określonym przedziale wysokości, wykrywa automatycznie kody alarmowe, pozwala na wpisanie do pamięci całej listy startów na dany dzień, o których informacja jest wyświetlana na 10 minut przed wpisanym czasem startu. Ponadto możliwe jest w ramach tego systemu przekazywanie zobrazowania na inny wskaźnik, do innego systemu radiolokacyjnego, śledzenie samolotów powyżej czy poniżej zadanej wysokości, śledzenie samolotów o wyselekcjonowanych numerach kodów itp.

System jest wyposażony w tzw. kursor — generowaną elektronicznie linię „doczepianą" do dowolnego miejsca ekranu i pozwalającą mierzyć odległość i kurs pomiędzy dwoma punktami.

Oprócz informacji otrzymywanych z powietrza czy też za pomocą radarów kontroler musi być również dobrze poinformowany o warunkach pogodowych na trasie, na lotnisku docelowym czyli po prostu musi być informacyjnym omnibusem. Nie wystarczy wiedzieć co się dzieje na własnym lotnisku czy nad terytorium Polski. Gdyby na przykład lotnisko docelowe było z jakichś powodów zamknięte to taki lot nie miałby po prostu sensu.

CO SŁYCHAĆ W ANKARZE

Obok stanowiska stoi zwykła drukarka dalekopisowa. Jest ona końcówka drugiego systemu komputerowego opartego na urządzeniu firmy **MITRA**, którego zadaniem jest przyjmowanie i nadawanie depesz na cały świat. Komputer pełni tu bardzo poważną rolę — jest odpowiedzialny za sortowanie, adresowanie I przesyłanie depesz do poszczególnych organów i służb ruchu lotniczego. Wyobraźmy sobie, że np. Turcja nadała na cały świat informację o rozpoczynających się za tydzień pracach polowych na lotnisku w Ankarze w związku z czym lotnisko będzie zamknięte. Taka informacja powinna być przekazana do Biura Od praw Załóg gdzie zgłaszają się wszyscy piloci przed planowanym lotem. Kontrolera interesują tylko informacje z ostatniej chwili. Resztę załatwia **Biuro Odpraw Załóg.**

Zupełnie inaczej wygląda sytuacja gdy jakieś lotnisko przysyła depeszę o zamknięciu zupełnie niespodziewanie. Informacja taka powinna jak najszybciej trafić do kontrolera,gdyż musi on ją przekazać do samolotów wykonujących już lot na to lotnisko, a ponadto może on jeszcze zatrzymać samoloty już gotowe do startu. System ten jest porównywalny do centralnego układu nerwowego — i jest nim w lotnictwie w rzeczywistości.

TRENING CZYNI MISTRZA

Na tym - rzecz jasna - nie kończy się zastosowanie komputerów w lotnictwie. W tym samym **Centrum Kontroli Ruchu Lotniczego** znajduje się też komputer **MERA**, spełniający rolę ... łoża tortur dla przyszłych kontrolerów radarowych. Łatwo się można domyślić, że chodzi tu o symulator radarowy, na którym muszą przejść odpowiednie przeszkolenie wszyscy ci, którzy myślą o fotelu przed ekranem radarowym. Oczywiście instruktorzy „dbają" o kursantów aby im się przypadkiem nie nudziło — a to nagle samolot po starcie zaczyna się palić, to znów w tym samym momencie dwa inne tracą łączność i na dodatek „siada" radar... i tak dalej i tak dalej. Możliwość symulowania „gorących" sytuacji jest tu oczywiście połączona z nauką odpowiedniego postępowania w takich wypadkach, co w prawdziwym ruchu pozwoli takie sytuacje rozwiązywać bez większych emocji.

SZYBKO, BEZBŁĘDNIE I OSZCZĘDNIE

W nowoczesnym lotnictwie nie obejdzie się także bez systemów kontrolnych. Taką rolę spełnia na warszawskim lotnisku komputer **HEWLETT PACKARD**, który służy do określania dokładności wskazań wszystkich naziemnych urządzeń radionawigacyjnych. Jest on zainstalowany w samolocie technicznym wraz z odpowiednią konsoletą; podczas pomiarów wyniki są zapisywane na dyskietkach i analizowane w zależności od potrzeby na ziemi lub w powietrzu. Tego rodzaju analiza pozwala na znaczne zmniejszenie kosztów związanych z drogimi oblotami technicznymi i oczywiście znacznie przyśpiesza lokalizację niesprawności czy ewentualnych uszkodzeń.

Drugi komputer tej samej firmy jest wykorzystywany do obliczeń związanych ze zużyciem paliwa — mam na myśli wszelkiego rodzaju wyliczenia trasowe. Jak wiadomo zużycie paliwa zależy nie tylko od typu samolotu i długości trasy; mają na to także wpływ wysokości lotu, warunki atmosferyczne, pora roku i wiele innych czynników.

ELEKTRONICZNA DRUKARNIA

Ostatnio warszawskie lotnisko wzbogaciło się o jeszcze jeden nowy system komputerowy oparty na maszynie firmy **IBM** — system pozwalający na redagowanie i wydawanie specjalistycznych publikacji lotniczych.

LOTNISKO BEZ LUDZI?

Bez komputerów nie ma żadnej przyszłości dla lotnictwa. Tylko komputery mogą dać sobie radę z uporządkowaniem i odpowiednim posortowaniem lawiny informacji i tylko komputery mogą zapewnić odpowiednią szybkość jej przesyłania oszczędzając cenny czas i wykonując za człowieka najbrudniejszą robotę. Dzięki temu ma on możliwość skupienia się nad tymi zadaniami, którym nie sprosta nawet najlepszy komputer. W niektórych krajach biura odpraw załóg to niewielki pokój z monitorem i klawiaturą; człowieka tam nie uświadczysz a jeżeli już, nie będzie to technik lub konserwator.

Nie ma się co łudzić — prędzej czy później komputerowa przyszłość czeka też polskie lotnictwo komunikacyjne. Na Okęciu pracuje już kilkanaście o ile nie kilkadziesiąt komputerów — początek wygląda więc bardzo obiecująco...

Klaudiusz Dybowski

BITY ZAMIAST CZCIONEK

Przy monitorach ekranowych pracują wyłącznie sympatyczne dziewczyny

Gdyby Gutenberg dwadzieścia lat temu zwiedzał przeciętną drukarnię gazetową, w pierwszej chwili złapałby się za głowę z powodu natłoku ludzi i urządzeń, hałasu oraz tempa maszyn drukarskich. Ale po pewnym czasie stwierdziłby z satysfakcją, że od pięciuset lat — gdy rozpowszechnił swój wynalazek — nie zmieniły się zasady druku wypukłego. Ale dwadzieścia lat temu nie było w drukarniach komputerów.

Właśnie „Bajtek", drukowany w **Prasowych Zakładach Graficznych RSW „Prasa-Książka-Ruch" w Ciechanowie**, powstaje przy pomocy najnowocześniejszych technik skomputeryzowanego fotoskładu.

Najpierw maszynopis (na przykład tego artykułu) trafia na stanowisko operatora monitora ekranowego (właściwie operatorki, gdyż na tych stanowiskach pracują prawie wyłącznie kobiety). W systemie SKIT pracuje pięc takich stanowisk. Jednostką sterującą jest system polskiej produkcji z pamięcią SP45DE. Na stanowiskach korektorki ekranowej odbywa się kodowanie tekstu oraz jego korekta. Nośnikami pamięci są 8-calowe dyski. Pod pojęciem „kodowanie" kryje się również zapisanie informacji jakim krojem pisma, wielkością czcionki ma być złożony tekst.

Teksty zakodowane na dyskach 8-calowych przenosi się na dyski 5 1/4 cala. wykorzystywane w CRTerminalu. który wyposażony jest w interface LCI (Linotype Communication Interface). To „przepisywanie" z dysku na dysk wynika z całkiem prozaicznych powodów — z różnorodności sprzętu, jakim dysponuje drukarnia. Od tej pory najważniejszym elementem w powstawaniu „Bajtka" będzie właśnie ta dyskietka. Na niej zostaną zapisane wszystkie informacje potrzebne do wydrukowania fragmentów czasopisma, a później całych stron czyli kolumn.

Ważnym etapem jest wydruk tekstu na drukarce TRD 32. Tekst, który powstanie na papierze, trafia do korekty, gdzie porównywany jest z maszynopisem oraz do redaktora technicznego.

Druk czy fotografia?

Po ostatecznej korekcie oraz wprowadzeniu uwag redaktora technicznego, dyskietka trafia do wy-swietlarki CRTronic 200. która wyposażona jest w dwa dowolnie programowane mikroprocesory. 704 K pamięci oraz dwie niezależne stacje dyskietek, każda 5 1/4 cala. Dyskietka formatowana jest na 320 K. Sercem urządzenia jest katodowa lampa elektronowa. Dlaczego sa aż dwie niezależne stacje dyskietek? Znowu wymóg tempa: w czasie, gdy na jednej dyskietce nanosi się ostateczne uwagi — dokonuje szlifu tekstu — z drugiej już odczytywany tekst jest wyświetlany przez lampę na błonie fotograficznej TF 32 Foton-Bydgoszcz. Szybkość naświetlana błony — ok. 200 tys. znaków na godzinę! Następnie klisze „montuje się" w gotowe kolumny.

Trzeba jeszcze wspomnieć o takim „drobiazgu" jak barwne fotografie. Każdy z podstawowych w drukarstwie kolorów trzeba przenieść na odpowiednią kliszę, aby podczas druku offsetowego, po nałożeniu poszczególnych barw podstawowych, otrzymać „gotowe wielobarwne zdjęcie". Tu znowu pojawia się komputer. Urządzenie Magnascan 510 rozkłada barwny slajd na poszczególne kolory podstawowe: żółty, czerwony, niebieski i czarny. Nie tylko „rozkłada", ale także zapisuje natężenie barwy i poszczególne z nich naświetla na błonie fotograficznej — podobnie jak tekst.

A jednak druk

Dopiero w tym momencie (gdy zmontowane są klisze z tekstem, zdjęciami, tabelami, listingami itp. — oczywiście dla poszczególnych kolorów osobne) pojawia się metal. Przy użyciu odpowiednich technik, to, co zostało na kliszy sfotografowane (cały czas jest to technika fotografii czarno-białej) przenosi się na metalowe matryce. One trafiają na bębny maszyny, rotacyjnej i pozostaje tylko „szycie i krojenie". Technika fotoskładu pomija więc pracochłonny i szkodliwy dla zdrowia etap odlewania tekstu w wiersze i później składania ich w „ramki" czyli kolumny. Przy korekcie w technikach tradycyjnych, aby poprawie jedną literę, trzeba odlewać na nowo cały wiersz. Pojawia się możliwość nowego błędu. Każdy kto pracował z edytorem tekstu, wie jak prosta staje się korekta przy pomocy komputera. Ale nie dosc tych udogodnień. W pewnym momencie druku okazuje się. że na kolumnie zabrakło miejsca. W technikach tradycyjnych trzeba więc na nowo odlewać tekst złożony mniejszą czcionka. W fotoskładzie — wydaje się komputerowi polecenie, aby ten sam tekst „przepisał" mniejszą czcionka. Czy ciechanowska drukarnia jest najnowocześniejsza w kraju? Dyrektor naczelny Jerzy Górski broni się przed takim określeniem.

— Lepiej powiedzieć: jedna z nowocześniejszych. W tym roku likwidujemy techniki typograficzne, będziemy rozszerzać skład komputerowy. Nastawiamy się na produkcję wielobarwną.

Dzis Zakłady drukują m.in. 1,2 mln popularnych wsrod bibliofilów „miniaturek": 8 mln książek, 160 mln losów loteryjnych — w tym także dziennikarskiej loterii „Błękitna" — druk losów jest dochodowy. I oczywiście czasopisma, wśród których „Bajtek" pod względem nakładów i objętości jest jednym z największych.

Sławomir Polak
Roman Wojciechowski

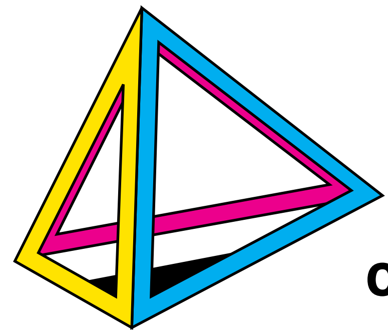
```
PH:TM 64 DO 78 ZAPISANO
ROZKAZ:KILL,1,64.
PZ:R=198.00mm; W=0; Y=293.50mm; X=0.00mm;
ROZKAZ:PUT
„BAJTEK" pod względem nakładu objęto-
ści jest jednym z największych kolorowych
czasopism drukowanych w Prasowych Za-
kładach Graficznych RSW „Prasa-Książka-
Ruch" w Ciechanowie.
```

Magnascan 510

Kolejny „Bajtek" trafia do czytelnika

WSTĘP DO PROGRAMOWANIA W JĘZYKU

PASCAL



cz. II

W pierwszym z niniejszego cyklu artykułów używaliśmy PASCAL-a do wykonywania prostych obliczeń. Obecnie zapoznamy się z całością możliwości obliczeniowych tego języka oraz z kompletem instrukcji, pozwalających na konstruowanie programu.

STANDARDOWE TYPY ZMIENNYCH

Jak już wiemy, w PASCAL-u istnieje obowiązek deklarowania nazw oraz typów używanych w programie zmiennych. Dotychczas posługiwaliśmy się wyłącznie zmiennymi typu REAL. Jest to jeden z czterech standardowych typów zmiennych w PASCAL-u. Wymieńmy je wszystkie:

integer
Typ ten odpowiada zbiorowi liczb całkowitych. Oczywiście nie jesteśmy w stanie reprezentować w komputerze wszystkich liczb całkowitych, lecz tylko ich skończony podzbiór.
Deklaracja zmiennej tego typu może wyglądać tak:
var LiczbaMinut: integer;
Jeżeli zachodzi potrzeba, możemy ów podzbiór liczb całkowitych zmniejszyć. Zadeklarujemy wówczas np.
var Procent: 0..100;
W powyższym przypadku zmienna Procent może przyjmować wartosci całkowite z przedziału 0..100.

real
Typ odpowiadający zbiorowi liczb rzeczywistych — z podobnymi, co poprzednio, zastrzeżeniami. Znamy już wygląd deklaracji takiej zmiennej: var Promien : real;

boolean
Dwuelementowy zbiór wartości logicznych: true (prawda) i false (fałsz). Zmienna tego typu, zadeklarowana jako np.
var Warunek : boolean; może być użyta np. w podstawieniach
Warunek :=true;
Warunek :=(2 = 3);
Warunek : Warunek and not Warunek;
char

Zbiór znaków kodu ASCII. Wypisywalny znak tego kodu oznacza się poprzez ujęcie go w pojedyncze apostrofy. Np. zadeklarowawszy
var Odpowiedz:char;
możemy użyć np. podstawienia
Odpowiedz := 's';
Podobnie jak w przypadku typu integer, również tutaj możemy dostępny zbiór znaków zawęzić poprzez okrojenie typu:
var DuzaLitera : 'A'..'Z';

Bardzo istotną regułą PASCAL-a jest zasada zgodności typów. Mówi ona o tym, że nie jest dozwolone nadawanie zmiennej określonego typu wartości, nie będącei elementem tego typu. Oznacza to, że np. zmiennej typu boolean nie możemy przypisać wartości całkowitej. JEDYNYM WYJĄTKIEM od tej zasady jest dopuszczenie podstawienia liczby całkowitej na zmienną typu real. W poniższym programie wszystkie użyte podstawienia są błędne.

```
var  Znak : char;
     N : integer;
     Iks : real;
     Warunek := boolean;
begin
     Znak :=  2;
     N := Iks;
     N := 1.0; Warunek:  'c';
     IKS :  not Warunek
end.
```

OPERACJE ARYTMETYCZNE

Wykonując obliczenia na liczbach rzeczywistych, mamy do dyspozycji cztery podstawowe działania arytmetyczne użyte już przez nas w podawanych przykładach:
+ — dodawanie
- — odejmowanie
* — mnożenie
/ — dzielenie

Trzy pierwsze działania mogą też być wykonywane dla liczb całkowitych. Zamiast dzielenia dla liczb tych wprowadzono operatory:
a div b — część całkowita ilorazu a przez b
a mod b — reszta z dzielenia a przez b.

Oprócz powyższych działań podstawowych PASCAL oferuje użytkownikowi gamę funkcji standardowych, pozwalających na wykonywanie obliczeń:
abs (x) — wartość absolutna (moduł) liczby x
sqr (x) — kwadrat liczby x
sin (x) — funkcja sinus
cos (x) — funkcja cosinus
arctan (x) — funkcja arcus tangens
exp (x) — ex
ln (x) — logarytm naturalny liczby x
sqrt (x) — pierwiastek kwadratowy liczby x
trunc (x) — część całkowita liczby. Funkcja ta pozwala na zmianę wartości typu real na wartość typu integer
round (x) — zaokrąglenie x do liczby całkowitej. Zastosowanie podobne, jak w przypadku funkcji trunc
odd (i) — funkcja wskazująca na parzystość liczby całkowitej i; przyjmuje wartość true dla i nieparzystych i false w przeciwnym przypadku

OPERACJE LOGICZNE

W odniesieniu do zmiennych i wyrażeń logicznych (typu boolean) możemy używać następujących operacji:
a and b — koniunkcja wartości wyrażeń a i b,
a or b — alternatywa wartości a i b,
not a — wartość przeciwna wartości a.

DZIAŁANIA NA ZMIENNYCH ZNAKOWYCH

W odniesieniu do elementów kodu ASCII możemy stosować następujące funkcje:
succ(x) — element typu następny względem x,
pred (x) — element typu poprzedni względem x,
ord (x) — numer znaku w kodzie ASCII. ,
chr (x) — znak kodu ASCII o numerze n.

BUDOWA ALGORYTMU

Poznaliśmy już pewną ilość instrukcji, pozwalającą na budowę prostych programów. Znamy już np instrukcję warunkową postaci
IF <warunek> **THEN** <instrukcja>;
IF <warunek> **THEN** <instrukcja> **ELSE** <instrukcja>;
gdzie <warunek> jest wyrażeniem typu boolean <instrukcja> oznacza natomiast instrukcję prostą lub złożoną. Przypomnijmy jeszcze, że pod pojęciem instrukcji złożonej rozumiemy ciąg instrukcji — być może również złożonych — ujęty w słowa **BEGIN...END.**
Przypuśćmy teraz, że w programie pascalowym występuje następujący ciąg instrukcji.

```
if x = 0 then write ('Niedobrze');
if x = 1 then write ('Średnio');
if x = 2 then write ('Dobrze');
```

W tym przypadku w zależności od wartości zmienne**j x** podejmowane są różne działania. Użyliśmy w tym celu wielokrotnie instrukcji warunkowej. Ten sam fragment da się jednak zapisać prościej:

```
case x of
   0 : write ('Niedobrze');
   1 : write ('Średnio');
   2 : write ('Dobrze');
end.
```

Pierwszą linię tego przykładu, **„case x of"**. moglibyśmy przetłumaczyć jako „wybierz x spośród:". Następne trzy wiersze zawierają listę możliwych wartości zmiennej x oraz instrukcji. Wykonywanych w przypadku, gdy wartość x jest równa umieszczone) przed dwukropkiem etykiecie. Konieczne jest w tym przypadku wymienienie wszystkich możliwych wartośći zmiennej **x**. W przeciwnym wypadku efekt wykonania całej instrukcji może być nieokreślony Podobnie jak w instrukcji warunkowej, tak i tutaj mamy jednak możliwość określenia instrukcji wykonywanej w przypadku, gdy wartość x nie odpowiada żadnej z liczb umieszczonych na liście. Zmodyfikujemy nasz fragment następująco:

case x of
0 : write('Niedobrze);
1 : write ('Średnio'):
2 : write ('Dobrze';
else write ('Poniżej krytyki'):
end

Zapisując podobną instrukcję wewnątrz programu, należy pamiętać o braku średnika przed słowem **ELSE** oraz o zakończeniu instrukcji **CASE** słowem **END** (któremu wszakże nie odpowiada słowo **BEGIN**, jak w pozostałych przypadkach).

PĘTLE

Poznane dotychczas elementy języka PASCAL jeszcze nie pozwalają nam na budowę pętli wewnątrz programu. Przedstawmy więc postacie pętli dostępne w tym języku.

Bodaj najczęściej używaną postacią pętli jest instrukcja:

WHILE <warunek> **DO** <instrukcja>;

W wolnym tłumaczeniu linia powyższa oznacza „dopóki zachodzi wymieniony warunek, wykonuj daną instrukcję". Oto prościutki przykład użycia:

```
i : =0;
while i < 10000 do
   begin
     writeln (i);
     i: = + 1;
end;
```

Jak łatwo zauważyć, wykonanie tego krótkiego fragmentu spowoduje wypisanie kolumny liczb od 0 do 9999. Składnia PASCAL-a dopuszcza również użycie instrukcji w postaci

REPEAT <instrukcja> **UNTIL** <warunek>;

(„powtarzaj daną instrukcję do chwili, gdy zajdzie wymieniony warunek"). Użyjmy instrukcji REPEAT w podobnym do poprzedniego przykładzie.

```
i:=0;
repeat
   writeln (i);
   i : = i + 1
until i > = 10000;
```

Słowa **REPEAT** i **UNTIL** stanowią w tym przypadku klamrę zamykającą instrukcję złożoną wewnątrz pętli; w związku z tym nie zachodzi potrzeba dodatkowego użycia **BEGIN..END.**

Efekt działania obu naszych programików będzie identyczny. Wbrew pozorom jednak instrukcje **WHILE** i **REPEAT** nie są równoważne. Zauważmy, że w przypadku **WHILE** warunek końca jest sprawdzany przed wykonaniem instrukcji w pętli, natomiast w przypadku **REPEAT** — po jej wykonaniu. Z tego powodu instrukcja wewnątrz pętli **REPEAT** jest zawsze wykonywana co najmniej raz. Ostatnią z dostępnych w PASCAL-u postaci pętli jest

FOR <zmienna> := <wartość pocz.>; **TO** <wartość końcowa> **DO** <instrukcja>;

Jest to dobrze znana użytkownikom BASIC-a pętla **FOR**.

Użyjemy jej w tym samym celu, co poprzednio **WHILE** i **REPEAT** — wypiszemy liczby całkowite od 0 do 9999.

for Liczba := 0 to 9999 do writeln (Liczba);

Zmienna indeksująca Liczba musi być typu integer. Zapis powyższy przetłumaczony na język potoczny jako „dla zmiennej Liczba przyjmującej wartości od 0 do 9999 wypisz jej wartość".

Wartość zmiennej indeksującej jest po każdym wykonaniu instrukcji w pętli zwiększana o 1. Możemy również uzyskać proces odwrotny — zmniejszanie tej zmiennej w każdym kroku o 1. Napiszemy wówczas:

for Liczba := 9999 downto 0 do writeln (Liczba);

PASCAL nie przewiduje jednak wykonywania pętli FOR z dowolnym przyrostem zmiennej indeksującej. W razie potrzeby łatwo jest się posłużyć instrukcją **WHILE** lub **REPEAT**

Cztery krótkie programy zamieszczone obok ilustrują użycie w praktyce omówionych konstrukcji programowych.

Marek Wyrwidąb

```
program Miniliczba:

{Program obliczający najmniejsza co do
modułu liczbę reprezentowalna
przez komputer}

const Zero = 0.0;

var Iks1, Iks2 : real;

begin
  Iks1 := 1;
  writeln ('Momencik...');
  while Iks1 <> Zero do
     begin
       Iks2 := Iks1;
       Iks1 := Iks1 / 2;
end;
  writeln (' Najmniejsza reprezentowalna ');
  writeln (' liczba wynosi: '. Iks2);
end.
```

```
program Pytanie;
{Jedno banalne pytanie z historii Polski }

var Rok: integer;

begin
  writeln (' W ktorym roku byla bitwa pod Grunwaldem? '):
  writeln (' Czekam na odpowiedz! ');
  writeln:

  repeat
     read (Rok):
     write (' - ');
     if Rok = 1410 then
       writeln (' Dobra odpowiedz! ')
     else
       writeln (' Bardzo zle! Musisz probowac dalej... ');
     until Rok = 1410

end.
```

```
program ASCII;

{Program wypisujacy tablice znaków kodu ASCII,
od znaku nr 32 do 255, przy uzyciu standartowej
funkcji chr }

const  RozmEkranu = 20;
       IloscKolumn = 5;

var Indeks : integer;
     Enter : char:

begin
  writeln ('TABELA ZNAKÓW KODU ASCII : ');
  writeln;
  for Indeks := 32 to 255 do
     begin
       write (Indeks. ' : '. chr (Indeks). ' ');
       if Indeks mod IloscKolumn = 0 then
         writeln:
       if Indeks mod (RozmEkranu * IloscKolumn) = 0 then
         begin
                writeln (' Wciśnij ENTER... ');
                read (Enter)
         end
       end
end.
```

```
Program Test;

{Przyklad realizacji testu komputerowego}

var Odpowiedz : char;

begin
  writeln (' Pvtanie testowe : ');
  writeln;
  writeln (' Co po polsku oznacza "RAM" ? ');
  writeln ('   A: " pamiec stala " ');
  writeln ('   B: " pamiec o losowvm dcsteoie " ');
  writeln ('   C: " pamiec wewnętrzna " ');
  writeln ('   D: " pamiec o swobodnym dostepie " ');
  writeln (' Wybierz odoowiedz: A, B, C lub B...  ');
  read (Odpowiedz);
  writeln;
  case Odpowiedz of
     'A'.'a',
     'C'.'c'  : writeln (' Bardzo zle ! ');
     'B'.'b'  : writeln (' Jest lepsza odpowiedz.. ');
     'D'.'d'  : writeln (' Swietnie ! ');
  else
     writeln (' Pardon ? ');
  end
end.
```

POLSKIE LITERY

NIE ma chyba drugiego komputera — przynajmniej z tych, popularnych w naszym kraju — na którym z taką łatwością moglibyśmy definiować własne znaki uzyskiwane z klawiatury. Dostępne w BASIC-u Amstrada komendy: **SYMBOL, SYMBOL AFTER, KEY, KEY DEF,** rozwiązują problem w dwóch, trzech linijkach. I właśnie dlatego, w każdym programie napisanym w Polsce litery takie jak: ą, ć, ę... uzyskiwane są przez naciśnięcie innych kluczy. A może warto pokusić się o zaproponowanie standardu?

Program POLSKIE LITERY nie powstał w naszej redakcji, nie został także napisany przez żadnego z naszych współpracowników. Jest on fragmentem programu POLNOTES krążącego już od pewnego czasu wśród użytkowników Amstrada. Niestety, nie znamy jego autora, a więc nie możemy mu oddać pełnej satysfakcji. Zapraszamy go jednak do współpracy z naszą redakcją.

Znaki polskiego alfabetu zostały w tym programie zdefiniowane na kluczach funkcyjnych czyli dodatkowej klawiaturze numerycznej, jednak bez konieczności rezygnacji z korzystania z tej ostatniej. Pomysł jest prosty — małe litery, specyficzne dla naszej pisowni, otrzymujemy naciskając klucze funkcyjne łącznie z kluczem **SHIFT** natomiast duże z **CONTROL**

Zasady działania programu nie musimy chyba wyjaśniać, gdyż wynika ona bezpośrednio z działania komend BASIC-a opisanych w każdej instrukcji obsługi. Przedstawiamy za to rozmieszczenie znaków na klawiaturze funkcyjnej.

(rp)

ROZMIESZCZENIE POLSKICH LITER PO ZMIANIE

sam klawisz	z SHIFT	z CTRL
↑	↑	↑
← □ →	← □ →	← □ →
↓	↓	↓
7 8 9	ą ć ę	Ą Ć Ę
4 5 6	ł ń ó	Ł Ń Ó
1 2 3	ś ż ź	Ś Ź Ż
0 . □	0 . □	0 . □

```
1000 SYMBOL AFTER 170
1005 SYMBOL 171,0,0,120,12,124,204,118,3
1010 SYMBOL 172,24,60,102,102,126,102,102,3
1020 SYMBOL 173,24,0,60,102,96,102,60,0
1030 SYMBOL 174,24,60,102,192,192,102,60,0
1040 SYMBOL 175,0,0,60,102,126,96,60,6
1050 SYMBOL 176,254,98,104,120,104,98,254,3
1060 SYMBOL 177,56,24,28,24,56,24,60,0
1070 SYMBOL 178,240,96,112,96,226,102,254,0
1080 SYMBOL 179,24,0,220,102,102,102,102,0
1090 SYMBOL 180,24,198,230,246,222,206,198,0
1100 SYMBOL 181,24,0,60,102,102,102,60,0
1110 SYMBOL 182,12,56,108,198,198,108,56,0
1120 SYMBOL 183,24,0,60,96,60,6,124,0
1130 SYMBOL 184,24,60,96,60,6,102,60,0
1140 SYMBOL 185,24,0,126,76,24,48,126,0
1150 SYMBOL 186,254,198,140,126,48,102,254,0
1160 SYMBOL 187,12,24,126,76,24,48,126,0
1170 SYMBOL 188,48,254,172,24,50,102,254,0
1220 KEY DEF 10,1,55,171,172
1230 KEY DEF 11,1,56,173,174
1240 KEY DEF 3,1,57,175,176
1250 KEY DEF 20,1,52,177,178
1260 KEY DEF 12,1,53,179,180
1270 KEY DEF 4,1,54,181,182
1280 KEY DEF 13,1,49,183,184
1290 KEY DEF 14,1,50,185,186
1300 KEY DEF 5,1,51,187,188
```

TROCHĘ RUCHU...

... i odrobina zabawy nie zaszkodzi. Nagrodą za pracowite „wstukanie" programu będzie nie tylko ruchoma reklama „Bajtka". Zmieniając tekst w liniach od 120 do 180 możemy również zareklamować własne umiejętności.

Program wykorzystuje możliwość programowania układu scalonego 6845, czyli kontrolera ekranu w Amstradzie. Wysokość tekstu może być zmieniona od normalnego wymiaru znaku, a więc jednej linii aż do 16-tu linii (patrz końcowe cyfry w liniach DATA).

Natomiast w liniach 330 do 560 zapisana jest w kodzie maszynowym procedura przesuwania liter na ekranie oraz programowanie koloru tła i napisów (linia 60). Kolory te (1 +RND*255) nie mają nic wspólnego z używanymi przez Basic, są to argumenty wprowadzane bezpośrednio do pamięci ekranu. Poeksperymentuj z nimi do czasu znalezienia odpowiadającej ci kombinacji.

Uwaga: maksymalna długość tekstu zależy od jego wielkości — im mniejsze litery, tym dłuższy tekst może być wyświetlany w jednej linii.

Wojciech Ziółek

```
10 ` Wedrujace litery
20 GOSUB 200:RESTORE
30 BORDER 0:INK 0,0
40 WHILE NOT asleep
50 READ a$,size:IF a$="END" THEN RESTORE:GOTO 50
60 CALL &A000,@a$,1+RND*255,1+RND*255,size
70 WEND
80 RESTORE
90 WEND
100 END
110 :
120 DATA "Czytaj Bajtka....",2
130 DATA "Czytaj Bajtka....",4
140 DATA "Czytaj Bajtka....",8
150 DATA "Czytaj Bajtka....",16
160 DATA "Czytaj Bajtka....",4
170 DATA "Tylko Bajtek nauczy cie poslugiwac sie
    komputerem         ",4
180 DATA "Jego autorzy dbaja o wysoki poziom
    prezentowanego materialu        ",16
190 DATA END,1
200 ` ----Store Machine Code----
210 MEMORY &9FFF:adress=&A000
220 RESTORE 330
230 FOR  i=1  TO 24
240 sum=0:READ code$,check$
250 FOR j=1 TO 21 STEP 2
260 byte=VAL("&"+MID$(code$,j,2))
270 POKE adress,byte
280 sum=sum+byte:adress=adress+1
290 NEXT
300 IF sum<>VAL("&"+check$)  THEN PRINT "Error
in data line";340+i*10
310 NEXT
320 RETURN
330 DATA a7c83d28243d280add7e00,3c2
340 DATA 3288a0dd23dd233d280add,4a6
350 DATA 7e003286a0dd23dd23dd7e,531
360 DATA 003280a0dd23dd23dd6e00,49d
370 DATA dd66017ea7c83209a12322,452
380 DATA 3ba0dd2a00003e01cd0ebc,3b8
390 DATA 219fc02266a02100002207,2f2
400 DATA a1dd7e00cda5bbcd06b911,5c6
410 DATA ffa0010800edb0cd09b906,4da
420 DATA 08f3110000c5cb40ca78a0,4be
430 DATA 0601cdcba010fbcddca006,599
440 DATA 0821ffa0c5e50effcb2638,5a8
450 DATA 020e000608626b71233600,1b5
460 DATA 2b7cc60867d29ba01150c0,50a
470 DATA 1910ee545de123c110d92a,4a0
480 DATA 66a0014f00a7ed42060836,370
490 DATA 007cc6086710f82a66a023,40c
500 DATA 2266a0c110a4dd232109a1,468
510 DATA 35c24ea0c9f5c506f5ed78,6c8
520 DATA 1f38fbed781f30fbc1f1c9,67c
530 DATA 2a07a123232207a1cb3ccb,3b4
540 DATA 1d0100bc3e0ced79047cc6,3d0
550 DATA 30ed79053e0ded79047ded,4ba
560 DATA 79c9000000000000000000,142
```



PAMIĘĆ POD LUPĄ

Podczas pracy na komputerze przydatna jest czasem możliwość „podglądania” zawartości pamięci lub jej modyfikacji. Wymagania takie spełnia przedstawiony tutaj prosty program, nazwany monitorem.

Dla łatwiejszego zrozumienia działania programu podstawowe jego bloki (podprogramy) opatrzone są krótkim komentarzem. W momencie uruchomienia programu blok „Inicjalizacja” definiuje tryb pracy monitora, kolory i okna tekstowe. Ustala reakcję programu na zatrzymanie klawiszem ESC przez odesłanie do podprogramu „Zatrzymanie”, a następnie przechodzi skokiem bezwarunkowym (linia 80) do podprogramu „Menu”.

W zależności od wybranej opcji (linie 390-420) uruchamiane zostają podprogramy: sprawdzanie pamięci, modyfikacja pamięci lub sprawdzanie pamięci z jednoczesnym wydrukiem na drukarce. Wybór opcji 3 (powrót do BASIC-a) powoduje wyczyszczenie ekranu (MOD 2) i zatrzymanie wykonywania programu (STOP).

Wciśnięcie klawisza **ESCAPE** podczas pracy powoduje zatrzymanie programu tylko do momentu wciśnięcia dowolnego innego klawisza. Przerwanie wykonywania programu i powrót do Menu uzyskuje się przez przytrzymanie przez chwilę klawisza M (patrz linie 460, 260, 600).

Spójrzmy jeszcze na linie 110, 120 i 130, w których dokonuje się dwoma różnymi metodami tablicowanie wydruku i porównajmy je z liniami 210, 220 i 230. Wnioski pozostawiamy Czytelnikom.

na podstawie Amstrad Magazine

Wojciech Ziółek

```
10  `*** Monitor ***
20  `*** Inicjalizacja ***
30 MODE 2:WINDOW #1,3,78,4,14:PEN #1,0:PAPER
   #1,1:CLS #1
40 WINDOW #2,5,50,16,17:WINDOW
   #3,3,78,19,23:PEN #3,0:PAPER #3,1:CLS #3
50 WINDOW #4,3,78,2,3:PEN #4,0:PAPER #4,1:CLS
   #4
60 LOCATE 22,1:PRINT "------Monitor kodu
   maszynowego------"
70 ON BREAK GOSUB 450
80 GOTO 310
90 `*** Sprawdzanie pamieci ***
100 INPUT #2,"Adres poczatkowy ";adr
110 CLS #4:PRINT #4,"         Adres pamieci
    Zawartosc pamieci"
120 PRINT #4,"      dziesietny     hex";"
    ";"dziesietna     hex       bin        chr$"
130 adr=ABS(adr):PRINT #1,TAB(10)adr;TAB(21)
    HEX$(adr,4);TAB(41)PEEK(adr);TAB(52)
    HEX$(PEEK(adr),4);TAB(60)
    BIN$(PEEK(adr),8);
140 IF PEEK(adr)>32 THEN PRINT #1,TAB(74)
    CHR$(PEEK(adr));:PRINT #1
150 IF adr=65535 THEN STOP ELSE adr=ABS(adr)+1
160 IF INKEY$="m" OR INKEY$="M" THEN 310
170 GOTO 130
180  `*** Modyfikacja pamieci ***
190 CLS #1:INPUT #2,"Od jakiego adresu chcesz
    zmieniac zawartosc pamieci? ",adr
200 IF adr>65535 THEN 190
210 CLS #4:PRINT #4,TAB(5)"Adres komorki
    pamieci";TAB(55)"Zawartosc pamieci"
220 PRINT #4,"dziesietny";TAB(22)"hex"
230 PRINT #1,TAB(2)adr;TAB(21)
    HEX$(adr,4);TAB(64);
240 INPUT #1,al$
250 IF al$=""  THEN 290
260 IF al$="m" OR al$="M" THEN 310
270 IF VAL(al$)>255 THEN 230
280 POKE  (adr),VAL(al$)
290 IF adr=65535 THEN END ELSE adr=adr+1
300 GOTO 230
310  `*** Menu ***
320 PRINT #3,"        --------------MENU-----
    ----------------- "
330 PRINT #3,"         1--->sprawdzanie pa-
    mieci"
340 PRINT #3,"         2--->modyfikacja pa-
    mieci"
350 PRINT #3,"         3--->powrot do Basic'a"
360 PRINT #3,"         4--->sprawdzanie pa-
    mieci z jednoczesnym wydrukiem"
370  `*** Sprawdzanie wybranej opcji ***
380 INPUT #2,"Co wybierasz z menu? ",a$
390  IF a$="1"  THEN 90
400  IF a$="2"  THEN  180
410  IF a$="3" THEN MODE 2:STOP
420  IF a$="4"  THEN 490
430 GOTO 380
440  `*** zatrzymanie ***
450 LOCATE 2,24:PRINT  "przerwa"
460 IF INKEY$=""  THEN 460
470 LOCATE 2,24:PRINT "
480 RETURN
490 `*** Sprawdzanie pamieci z jednoczesnym
    wydrukiem ***
500 INPUT #2,"Adres poczatkowy ";adr
510 CLS #1:PRINT #8,"----poczatek---> ";adr
520 CLS #4:PRINT #4,"         Adres pamieci
    Zawartosc pamieci"
530 PRINT #4,"      dziesietny     hex";"
    ";"dziesietna     hex       bin        chr$"
540 PRINT #8,"         Adres pamieci
    Zawartosc pamieci"
550 PRINT #8,"      dziesietny     hex";"
    ";"dziesietna     hex       bin        chr$"
560 adr=ABS(adr):PRINT #1,TAB(10)adr;TAB(21)
    HEX$(adr,4);TAB(41)PEEK(adr):TAB(52)
    HEX$(PEEK(adr),4);TAB(60)
    BIN$(PEEK(adr),8);
570 IF PEEK(adr)>32 THEN PRINT #1,TAB(74)
    CHR$(PEEK(adr));:PRINT #1
580 PRINT #8,TAB(10)adr;TAB(21)
    HEX$(adr,4);TAB(41)PEEK(adr);TAB(52)
    HEX$(PEEK(adr),4);TAB(60)
    BIN$(PEEK(adr),8);
590 IF adr=65535 THEN STOP ELSE adr=ABS(adr)+1
600 IF INKEY$="m" OR INKEY$="M" THEN 310
610 GOTO 560
```

RENUMERACJA

Przedstawiam program ZX-Spectrum, służący do automatycznej renumeracji linii. Z wielu innych programów tego typu wyróżnia go:

— szybkość działania,

— początek renumeracji rozpoczyna się od linii wskazywanej przez kursor programowy (co go różni od innych tego typu programów) — bardzo przydatne,

— wartość skoku linii może mieć dowolną liczbę — wpisuje się ją sekwencją:

POKE 60001 ,SKOK-256xINT (SKOK/256) POKE 60002,INT (SKOK/256)

— jest relokowalny,

— przyjmując wartość skoku=16384 (POKE 60002,64), linie znajdujące się za kursorem zostają usunięte z ekranu (nie z komputera) i nie biorą udziału w programie. Przywrócenie ich z powrotem następuje przy wartości skoku mniejszej od 9999. Jest to bardzo przydatne przy pisaniu długich programów (linie te wówczas nie przeszkadzają).

Ze względu na długość tego programu (RANDOMIZE USR 60000) należy uważać, aby kursor programu nie był „ukryty".

Andrzej Dziarnowski

```
10 FOR n=60000 TO
   60053
20 READ a: POKE n,a
30 NEXT n
40 DATA 1, 10, 0, 42,
   83, 92, 86, 35, 94,
   229, 235, 237, 91,
   73, 92, 237, 82,
   225, 48, 8, 35, 94,
   35, 86, 35, 25, 24,
   234, 43, 114, 35,
   115, 35, 215, 94,
   35, 86, 55, 25,
   237, 91, 75, 92,
   229, 167, 237, 82,
   209, 225, 208, 9,
   235, 24, 231
```

KŁOPOTY Z PAMIĘCIĄ

Powszechnie znane są kłopoty w ZX SPECTRUM związane z uszkodzeniem pamięci RAM 4116. Najczęstszą przyczyną tego uszkodzenia jest zła praca przetwornicy napięcia.

Gdy coś takiego się już zdarzy — czego nikomu nie życzę — można przy wymianie pamięci wykorzystać układy 4116 zasilane pojedynczym napięciem +5V. W polskim katalogu produkowanych układów scalonych, pamięci takie są zamieszczone pod oznaczeniem MCY 7161. Wymieniając pamięci na zasilane pojedynczym napięciem, należy dokonać drobnych zmian w połączeniu końcówek układów, które wynikają ze zmiany zasilania. Dla układu 4116 w wersji trójnapięciowej (montowane w standardowej wersji ZX SPECTRUM) zasilanie podane jest na:

- 5V — końcówka 1 + 12V —

SŁUPKI

Program „Wykres" służy do przedstawiania danych liczbowych w postaci pseudotrójwymiarowych słupków.

Na początku program pyta się o tytuł wykresu. Można wpisać dowolny tekst, jednak tylko pierwsze 32 znaki będą przyjęte. Następnie program pyta się o rodzaj skalowania. Jeśli wybrane zostanie skalowanie automatyczne wówczas wykres będzie skalowany według największego słupka, jeśli nie — program spyta o wartość według której odbędzie się skalowanie.

Teraz należy podać ilość słupków (maksymalnie — 16) oraz określić wartość, kolor i nazwę każdego z nich. Trzeba pamiętać, że przy skalowaniu nieautomatycznym wartości słupków nie mogą być większe od wartości według której odbywa się skalowanie.

Po zdefiniowaniu wszystkich słupków, na ekranie rozpocznie się rysowanie. Po jego zakończeniu użytkownik ma następujące opcje.

Klawisz „D" — włączenie/wyłączenie opisu liczbowego wykresu

Klawisz „L" — wyświetlenie legendy wykresu. Naciśnięcie klawisza ponownie spowoduje powrót do wykresu

Klawisz „W" — wydruk ekranu na drukarce (standardowa instrukcja „COPY"

Klawisz „N" — nowy wykres.

80-340 wczytanie danych i parametrów wykresu
350-420 skalowanie
430-550 rysowanie wykresu
560-610 czytanie klawiatury
620-730 podprogram rysujący słupki
740-805 podprogram wprowadzający kod maszynowy do szybkiego przełączania ekranów
810-930 podprogram rysujący opis liczbowy
940-960 podprogram rysujący legendę.

Michał Szuniewicz

```
10 BORDER 0: PAPER 0: INK 7
20 POKE 23658,8
30 LET X=0: LET Y=0: LET Q=0
40 LET W=0: LET EK=150: LET SV=0: LET
   C$="": LET Q$=""
50 LET D=10: LET G=4: LET L=8
60 LET LEG=0: CLS
70 GO SUB 740
80 FOR T=0 TO 31: LET C$=C$+CHR$ 32: NEXT T
90 PRINT AT 21,0;"TYTUL:"
100 INPUT LINE T$
110 IF LEN T$>32 THEN LET T$=T$(1 TO 32)
120 CLS
130 PRINT AT 21,0;"SKALOWANIE AUTOMATYCZNE
    (T/N) ?"
140 INPUT LINE A$
150 CLS
160 LET SA=1
170 IF A$="N" OR A$="n" THEN PRINT AT
    21,0;"DO JAKIEJ WARTOSCI SKALOWAC ?":
    INPUT S: LET SA=0: CLS
180 PRINT AT 21,0;"ILOSC SLUPKOW (MAX-
    16):": INPUT IS: CLS
190 IF IS>16 THEN BEEP .1,30: GO TO 180
200 DIM W(IS): DIM K(IS): DIM P(IS): DIM
    N$(IS,12)
210 FOR T=1 TO IS
220 PRINT AT 21,0;"WARTOSC SLUPKA #";T:
    INPUT W(T): LET W(T)=ABS W(T)
230 IF SA=1 THEN LET S=W(T)
240 IF SA=0 AND W(T)>S THEN BEEP .1,30:
    PRINT AT 21,0;"WARTOSC WIEKSZA OD
    SKALI": PAUSE 100: PRINT AT 21,0;C$: GO
    TO 220
250 PRINT AT 21,0;"KOLOR SLUPKA #";T;" (1-
    7)": INPUT K(T): PRINT AT 21,0;C$
260 IF K(T)<1 OR K(T)>7 THEN BEEP .1,30:
    PRINT AT 21,0;"NIE MA TAKIEGO KOLORU":
    PAUSE 100: PRINT AT 21,0;C$: GO TO 250
270 PRINT AT 21,0;"NAZWA SLUPKA #";T: INPUT
    LINE N$(T)
280 PRINT AT T,0;T;")";AT T,3; INK
    K(T);CHR$ 143; INK 7;N$(T);" =";W(T)
290 NEXT T
300 PRINT AT 21,0;C$
310 POKE 40032,196: POKE 40035,27:
    RANDOMIZE USR 40027
320 PRINT AT 21,0;"LISTA O.K (T /N) ?":
    INPUT LINE A$
330 IF A$="N" OR A$="n" THEN CLS : GO TO
    210
340 CLS
350 IF SA=0 THEN GO TO 400
360 FOR T=1 TO IS
370 IF W(T)>W THEN LET W=W(T)
380 NEXT T
390 LET S=W
400 FOR T=1 TO IS
410 LET P(T)=INT (W(T)*EK/S)
420 NEXT T
430 GO SUB 810
440 FOR T=1 TO IS
450 INK K(T)
460 FOR Y=0 TO P(T)
470 GO SUB 620
480 NEXT Y
490 IF P(T)<7 THEN GO SUB 690
500 OVER 1: GO SUB 670: OVER 0
510 PLOT OVER 1;X+D+G,Y+L-1: DRAW OVER
    1;0,-(P(T)+1)
520 LET X=X+16
530 NEXT T
540 PRINT AT 0,0; BRIGHT 1; INK 7; INVERSE
    1;T$
550 POKE 40032,227: POKE 40035,27:
    RANDOMIZE USR 40027
560 LET A$=INKEY$
570 IF (A$="D" OR A$="d") AND LEG=0 THEN
    RANDOMIZE USR 40000
580 IF A$="W" OR A$="w" THEN COPY
590 IF A$="N" OR A$="n" THEN RUN
600 IF A$="L" OR A$="l" THEN GO SUB 940
610 GO TO 560
620 PLOT X,Y: DRAW D,0
630 DRAW G,L: DRAW -D,0
640 DRAW -G,-L
650 PLOT OVER 1;X+D,Y
660 RETURN
```

KŁOPOTY Z PAMIĘCIĄ

końcówka 8 + 5V — końcówka 9

0V — końcówka 16

Natomiast dla wersji jednonapięciowej (np. MCY 7161) zasilanie| podane jest na:

X — końcówka 1 +5V— końcówka 8 X — końcówka 9 0V — końcówka 16

UWAGA: X oznacza końcówkę! nie podłączoną.

I jeszcze jedna bardzo ważna sprawa. Gdy to tylko jest możliwe, kasowanie pamięci mikrokomputera powinno odbywać się rozkazem RANDOMIZE USR 9 a nie wyciąganiem wtyczki zasilacza (dotyczy to ZX SPECTRUM bez przycisku RESET). W sytuacji gdy trzeba ją wyjąć, nie należy wkładać natychmiast po wyjęciu, lecz odczekać kilka sekund, aby niej spowodować nieprawidłowej pracy przetwornicy napięcia.

Konrad Fedyna

```
670 PLOT X+D+6,Y+L: DRAW -6,-L: DRAW -D,0:
    PLOT X+D+G,Y+L
680 RETURN
690 LET Q=X
700 FOR U=P(T) TO P(T)+L
710 PLOT Q,U: DRAW D,0
720 LET Q=Q+.5
730 NEXT U: PLOT OVER 1;X+D,Y-1: RETURN
740 FOR T=40000 TO 40038
750 READ V: POKE T,V
760 LET SV=SV+V
770 NEXT T
780 IF SV<>3229 THEN PRINT FLASH 1;"BLAD W
    LINI 800 LUB 805": STOP
790 RETURN
800 DATA 33,0,64,17,0,161,1,0,24,26,197,70,
    168,119,193,19,35,11,62,0,184,32,242,18
    5,32,239,201
805 DATA 33,0,64,17,0,161,1,0,24,237,176,2
    01
810 INK 0
820 FOR T=1 TO IS
830 LET Q$=Q$+(N$(T))(1 TO 1)+CHR$ 32
840 LET W$=STR$ W(T)
850 FOR I=1 TO LEN W$
860 LET J$=W$(I TO I)
870 IF P(T)<8*(LEN W$+2) THEN PRINT AT 21-
    INT (P(T)/8+2)-(LEN W$-I),2*T-2;J$
880 IF P(T)>-8*(LEN W$+2) THEN PRINT AT 21-
    INT (P(T)/8)+I,2*T-2;J$
890 NEXT I
900 NEXT T
910 PRINT #0,Q$
920 POKE 40032,161: POKE 40035,24:
    RANDOMIZE USR 40027
930 INK 7: CLS : RETURN
940 IF LEG=0 THEN POKE 40029,196: POKE
    40032,64: POKE 40035,27: RANDOMIZE USR
    40027: LET LEG=1: RETURN
950 IF LEG=1 THEN POKE 40029,227: POKE
    40032,64: POKE 40035,27: RANDOMIZE USR
    40027: LET LEG=0
960 RETURN
```

MASTER TOOLKIT

ZX SPECTURM posiada nie najlepsze oprogramowanie interpretera BASIC i dlatego powstało kilka programów tzw. „protez", które po wczytaniu do pamięci mikrokomputera rozszerzają jego możliwości. Przykładem takiego programu jest MASTER TOOLKIT firmy OCP. Program umożliwia wygodna obróbkę nowo pisanego programu poprzez danie do dyspozycji programiście szeregu nowych funkcji, które normalnie nie istnieją w ZX SPECTRUM.

Ładowanie programu odbywa się rozkazem: **LOAD „" <ENTER>**. „Toolkit" musi być załadowany przed programem w języku BASIC, który może być wprowadzany z klawiatury lub załadowany z taśmy za pomocą komendy **„LOAD"** lub **„MERGE"**. „Toolkit" wymusza przejście do trybu **INTERRUPT 2**. Jeżeli zostanie użyta komenda **„RUN"** to program przechodzi w tryb **INTERRUPT 1**, w którym „Toolkit" nie wykonuje swoich funkcji. Ponowne wznowienie pracy programu „Toolkit" i przejście do trybu **INTERRUPT 2** następuje po wykonaniu komendy: **RADOMIZE USR 65533**.

Wywołanie programu „Toolkit" odbywa się w normalnym trybie edycyjnym (tzn. gdy miga kursor w lewym dolnym rogu ekranu), poprzez równoczesne naciśnięcie klawiszy **SYMBOL SHIFT** i **SPACE** Wówczas ukazuje się nazwa programu i notatka o prawach autorskich. Rozkazy programu „Toolkit" mogą być wprowadzane za pomocą małych lub dużych liter.

Powrót do BASIC-u z programu „Toolkit" nastąpi po komendzie: **b <ENTER>**.

Komendy „Toolkita" można korygować w normalny sposób, tzn. za pomocą klawiszy kursora i **DELETE**, zaś ekran oczyszczać za pomocą rozkazu **CLS**.

Czasami zachodzi potrzeba wprowadzenia w programie „Toolkit" słowa kluczowego języka **BASIC** (np. przy przeszukiwaniu tekstu). Można tego dokonać w dwojaki sposób:

a) Przed wymaganym słowem kluczowym należy wprowadzić słowo **THEN** oraz wymagane słowo kluczowe, następnie przesunąć kursor przed słowo **THEN** i skasować je klawiszem **DELETE**.

b) Nacisnąć równocześnie klawisz **SPACE** i klawisz **K**, wtedy pojawi się migający kursor **K**, pozwalający wprowadzić wymagane słowo kluczowe.

UWAGA:

KOMENDY „TOOLKITA" MUSZĄ BYĆ WPROWADZANE LITERA PO LITERZE.

W podanych niżej opisach komend „Toolkita" operuje się słowem „zakres" oznaczającym zakres numerów linii programu w języku BASIC, w którym „Toolkit" ma wypełnić jakieś funkcje. A oto możliwe formy „zakresu":

n1 – n2 – od linii o numerze ni1 do linii n2;
n1 – – od linii n1 do końca programu;
– **n2** – od początku programu do linii n2 włącznie,
n1 – **n1** – tylko linia n1;
– – cały program w języku BASIC.

PODSTAWOWE KOMENDY PROGRAMU TOOLKIT

UWAGA: Pomiędzy komendą „Toolkita" a jej parametrami musi występować spacja, zaś wszelkie komendy muszą się kończyć naciśnięciem klawisza **ENTER**

cl g:m:s – ustawienie zegara na godzinę **g**, minutę **m** i sekundę **s**.

cl on – włączenie zegara i wyświetlenie czasu podanego wyżej.

cl off – wyłączenie wyświetlania czasu, nie powodujące zatrzymania zegara.

cl – wyświetlenie na ekranie bieżącego czasu.

UWAGA: Zegar przestaje pracować na okres działania rozkazu typu **LOAD, MERGE, VERIFY, BEEP**. To samo dotyczy funkcji „alarm", opisanej poniżej.

al g:m:s – ustawienie alarmu dźwiękowego na godzinę **g,** minutę **m** i sekundę **s.**

al on – włączenie gotowości alarmu.

al off – wyłączenie gotowości alarmu.

al – wyświetlenie czasu alarmu, powoduje likwidację sygnału alarmowego jeśli jest on wysyłany.

b – powrót do systemu BASIC.

l zakres – listowanie linii występujących w podanym zakresie, przy czym jeśli listing liczy więcej niż 22 wiersze to pojawia się napis „scroll?", na który należy odpowiedzieć tak samo jak w systemie BASIC.

mer n1, n2 – połączenie linii **n1** i linii **n2** w jedną linię o numerze n1; np. dla programu:

10 FOR a = 1 TO 10
20 NEXT a

komenda **mer 10, 20** powoduje utworzenie linii:

10 FOR a = 1 TO 10:NEXT a

zaś komenda **mer 20,10** tworzy linię następującą:

20 NEXT a:FOR a = 1 TO 10

mer n, n – zdublowanie linii o numerze n: np. dla linii:

15 PRINT q

po komendzie **mer 15,15** otrzymujemy:

15 PRINT q:PRINT q

del zakres – skasowanie linii w podanym zakresie. Kiedy podany jest pusty zakres, to kasowany jest cały program w języku BASIC. Po komendzie del pojawia się pytanie:

DELETE WHOLE? Y/N

na które odpowiedź **„Y"** powoduje jej wykonanie zaś odpowiedź **„N"** jej unieważnienie.

mov zakres,n,s – przeniesienie linii z danego zakresu w nowy, rozpoczynający się od numeru **n i o kolejnych numerach n+s, n+2s, n+3s...** Przy wykonywaniu tej komendy kasowane są linie w pierwotnym zakresie. W przypad-

ku braku miejsca na linie w nowym zakresie komenda nie jest wykonywana i pojawia się komunikat:
NO ROOM BETWEEN LINES

cop zakres,n,s – skopiownie linii z podanego zakresu w zakres nowy, rozpoczynający się od linii o numerze **n** i dalszych numerach **n+s, n+2s, n+3s...**
Jeśli numery kopiowanych linii pokrywają się z numerami linii już istniejących to komenda nie jest wykonywana i wyświetlany jest komunikat:
OUT OF MEMORY

f zakres \t – szukanie podanego tekstu **„t"** w danym zakresie (find), a po jego znalezieniu wyświetlenie numerów linii i instrukcji w liniach gdzie ten tekst występuje. Np. dla następującego programu:

10 FOR a=1 TO 10
20 PRINT a,a*a
30 NEXT a
40 PRINT
50 FOR a = 1 TO 10:PRINT a, SQR a:NEXT a

po rozkazie **f 10-50 PRINT** otrzymuje się wydruk: **20:1 40:1 50:2**

fl zakres \t – szukanie tekstu **„t"** w podanym zakresie i listing linii, w których ten tekst występuje (find & listing). Dla programu podanego powyżej po komendzie **fl 10-50 PRINT** otrzymuje się:

20 PRINT a, a*a
40 PRINT
50 FOR a = 1 TO 10:PRINT a, SQR a:NEXT a

s zakres tl \ t2 – zastąpienie w danym zakresie tekstu **„t1"** przez tekst **„t2"**

s zakres \t – usunięcie zadanego tekstu **„t"** w podanym zakresie numerów linii.

syntax zakres – kontrola składni w danym zakresie i ewentualna jej korekta.

fr – wyświetlenie komunikatu o długości obszaru wolnej pamięci w postaci np.
32741 BYTES FREE

map – przedstawienie w postaci graficznej mapy pamięci z podaniem adresów wskazywanych przez zmienne systemowe **PROG, VARS.....UDG.**

ramtop zakres – zmiana wartości zmiennej RAMTOP bez kasowania zmiennych w programie.
UWAGA: Zmienna **RAMTOP** nie może podawać adresu mieszczącego się w obszarze Toolkita oraz obszarze stosu kalkulatora (**STKEND** + 200 bajtów).

squash – usunięcie mniej ważnej części „Toolkita" **(map, cl, al)** o długości około 500 bajtów w celu ewentualnego umieszczenia w tym miejscu programu własnego.
UWAGA: jeśli po komendzie **„squash"** zostanie później podana komenda dotycząca nieistniejącej już części „Toolkita" to pojawi się komunikat.
INVALID COMMAND

p on – skierowanie wyjścia generowanego przez „ Toolkit" na drukarkę.

p off – skierowanie wyjścia generowanego przez „Toolkit" na ekran.
UWAGA: komunikaty o błędach (**INVALID COMMAND. NO ROOM BETWEEN LINES** itp.) występują tylko na ekranie.

lc zakres – zmiana wielkich liter (upper case) na małe (lower case) w podanym zakresie.

uc zakres – zmiana małych liter na wielkie w podanym zakresie.

xc zakres – zamiana wielkich liter na małe i małych na wielkie w podanym zakresie.

remkill zakres – usunięcie z podanego zakresu wszystkich instrukcji REM.

pack zakres – minimalizacja długości programu w podanym zakresie poprzez „upakowanie" do każdej linii możliwie dużej liczby instrukcji (do 127). Np. w przypadku programu

10 GO TO 60
20 FOR a = 1 TO 10
30 PRINT a
40 NEXT a
50 RETURN
60 PRINT „hello"
70 GO SUB 20
80 STOP

po komendzie **pack 10-80** otrzymamy następujący program:

10 GO TO 60
20 FOR a=1 TO 10:PRINT a:NEXT a: RETURN
60 PRINT „hello" :GO SUB 20:STOP

UWAGA: linia 60 w powyższym przykładzie nie może być „upakowana" do linii 20, gdyż nie można byłoby wykonać skoku wskazanego w linii 10.

compress zakres – usunięcie z zadanego zakresu wszystkich instrukcji REM, minimalizacja liczby linii w tym zakresie oraz zastąpienie wszystkich stałych numerycznych na wydruku zerami (binarna postać tych stałych nie zmienia się). Jest to największa mo żliwa kompresja programu, jego długość zmniejsza się wówczas o około 20%.

r zakres,n,s – renumeracja linii w podanym zakresie (n jest numerem pierwszej linii w danym zakresie po renumeracji, s jest skokiem numerów linii). Jeśli parametry n i s nie zostaną podane to „Toolkit" przyjmuje n = 10 oraz s = 10.
UWAGA: gdy nowe numery linii są większe od 9999 lub pokrywają się z numerami linii już istniejących to renumeracja nie jest wykonywana.

v – wyświetlenie nazw wszystkich zmiennych zdefiniowanych w programie wraz z ich wartościami.

v litera – wyświetlenie wartości zmiennych, których nazwy rozpoczynają się od podanej litery.

cls – oczyszczenie ekranu (podobnie jak komenda CLS w języku BASIC).

tr zakres,op,n:i – krokowe wykonanie programu w danym zakresie z opóźnieniem op/50 sekund pomiędzy instrukcjami. Śledzenie wykonywania programu odbywa się poprzez skok do instrukcji **„i"**w linii **„n"**.

error n:i – przekazanie sterowania do instrukcji **„i"** w linii **„n"** w przypadku wystąpienia błędu.

error – wyłączenie śledzenia błędów.

break n:i – wyłączenie systemowego śledzenia przerwań i przekazanie sterowania do instrukcji **„i"** w linii **„n"** po naciśnięciu klawisza BREAK.

break – wyłączenie obsługi przerwań.

k n – wyświetlenie łańcucha podstawionego pod klawisz liczbowy **n (n = 0 – 9).**

k – wyświetlenie wszystkich łańcuchów podstawionych pod klawisze numeryczne.

k n\str – podstawienie pod klawisz liczbowy **n** łańcucha **str**.

UWAGA:
1. Wywołanie łańcucha „str" następuje po równoczesnym naciśnięciu klawisza liczbowego n i klawisza SPACE.
2. Pionowa linia (SYMBOL SHIFT S w trybie E) może być używana w łańcuchu jako ENTER.
3. Znak „\" (backlash) musi być pisany w łańcuchu podwójnie, jeśli ma być drukowany a nie traktowany przez „Toolkit" jako ogranicznik.

Konrad Fedyna
Zygmunt Wereszczyński

UŚMIECHNIJ SIĘ!

TEST
NA JASNOWIDZENIE

Program ten jest adresowany do wszystkich, którzy mniej lub bardziej na poważnie biorą zjawiska spostrzegania pozazmysłowego. Pozwala na przetestowanie możliwości w tym zakresie. Nie należy tego brać na serio, gdyż nikt jeszcze nie stwierdził czy komputerowi można czytać w „myślach".

Działanie tego programu jest bardzo proste. Ponieważ starałem się, aby zawarte w nim komentarze były jak najpełniejsze, opiszę tutaj tylko ideę działania.

Na początku program generuje 25 liczb losowych. Umieszcza je w tablicy l (25) i przechodzi do następnej pętli. Tam pobiera od maksymalnie skupionego użytkownika odgadywane liczby. Jeżeli testowany odgadnie, zwiększa się liczba jego punktów. Na koniec komputer podaje wynik w procentach opatrzony krótkim komentarzem.

Użytkownik może wzbogacić program o efekty dźwiękowe, wykres, itd. Wydruk listingu pochodzi z komputera SINCLAIR ZX Spectrum ale z łatwością może być dostosowany do każdego innego komputera. Pomysł programu zaczerpnąłem z pisma „Atari User".

Grzegorz Zatryb (lat 17)

```
10 BORDER 0: PAPER 0: INK 7: CLS
20 DIM l(25): REM Tablica z liczbami
   losowymi
30 LET pkt=0: REM Licznik punktow
40 REM Ta petla tworzy  liczby losowe
50 FOR f=1 TO 25
60 LET l(f)=INT (RND*5)+1
70 NEXT f
80 PRINT AT 11,10;"1   2   3   4   5": REM
   Wydruk  liczb odgadywanych
90 REM Ta petla sprawdza zgodnosc  liczb
   pobranych w linii 110 z liczbami  w
   tablicy  l(25)
100 FOR f=1 TO 25
110 INPUT "Podaj liczbe :";s
120 IF s=l(f) THEN LET pkt=pkt+1: REM
    Dodawanie punktow
130 NEXT f
140 REM  Wyniki
150 LET proc=pkt*4: REM Zamiana punktow na
    procenty
160 CLS
170 PRINT AT 2,1;"Uzyskales  ";proc;" %."
180 REM Te  linie odsylaja program do
    odpowiedniej  linii  w zaleznosci  od
    ilosci  procent
190 IF proc<50 THEN GO SUB 230
200 IF proc>=50 AND proc<90 THEN GO SUB 250
210 IF proc>=90 THEN GO SUB 270
220 STOP : REM A teraz wyniki  testu
230 PRINT AT 3,1;"Niestety nie posiadasz
    zdolno-";AT 4,1;"sci jasnowidzenia."
240 RETURN
250 PRINT AT 3,1;"Twoje zdolnosci
    parapsycholo-";AT 4,1;"giczne
    przewyzszaja norme. Masz";AT
    5,1;"zadatki  na  jasnowidza."
260 RETURN
270 PRINT AT 3,1;"Twoje mozliwosci sa
    imponuja-";AT 4,1;"ce. Jestes dobrym
    jasnowidzem."
 280 RETURN
```



PRZEKLĘTE RACHUNKI

Wymagają skupienia i dokładności. Pochłaniają wiele czasu i niestety nie dają żadnej satysfakcji bo sa praca zupełnie bezmyślna. Znają ten problem wszyscy ludzie, którzy co miesiąc sumują długie kolumny liczb w planach, kosztorysach czy sprawozdaniach.

Mógłby ich w tej ciężkiej (naprawdę, ciężkiej!) pracy zastąpić komputer, ale to nie takie proste. Każdy z nich liczy coś trochę innego, więc każdy potrzebowałby innego programu. Na dodatek schematy obliczeń czasami się zmieniają, więc i programy trzeba by modyfikować. Koszty rosną — zastosowanie komputera staje się niezbyt opłacalne. I jeszcze aspekt psychologiczny. Ludzie nie bardzo rozumieją sposób pracy maszyny, trudno im się przestawić, bo są od wielu lat przyzwyczajeni do swoich formularzy i tabelek.

A właśnie, tabelek. Może więc zrobić program, który postawi do dyspozycji użytkownika tabelkę. Musi to być tabelka uniwersalna, po to żeby mogła służyć wszystkim. Dlatego, gdy po raz pierwszy pojawia się na ekranie komputera ma tylko ponumerowane wiersze i kolumny. Użytkownik sam będzie je nazywał zgodnie ze swoimi potrzebami i on określi, co znajdzie się w poszczególnych rubrykach, czy komórkach. Komórka (ang. „cell”) to miejsce, przeznaczone na jedną daną — liczbę lub tekst, powstałe na przecięciu wiersza i kolumny. W razie potrzeby skorzystania z danej wpisanej w komórkę bardzo tatwo się do niej odwołać — wystarczy podać numery wiersza i kolumny.

Chcielibyśmy bardzo, żeby nasza tabelka była „rozumna”. Na przykład, gdy wpiszemy do niej dane, to mogłaby policzyć i wpisać do odpowiednich komórek sumy wierszy i kolumn. Albo obliczyć średnią arytmetyczną, czy też policzyć ile liczb jest wpisanych w wierszach między siódmym a jedenastym. Pomysłowość ludzi przy stwarzaniu sobie najdziwniejszych potrzeb jest niezwykła. A przecież chcieliśmy mieć tabelkę dobrą dla wszystkich. Nie na więc innego wyjścia, każdy będzie musiał sam ustalać co ma obliczać jego tabelka. A jak mu się zmienią potrzeby, to sobie sam zmodyfikuje tabelce.

Jak to rozwiązać od strony technicznej (pamiętajny, że nie żądamy od ludzi umiejętności programo wania)? Otóż, do komórki niekoniecznie trzeba wpisywać liczbę, czy tekst. Zamiast tego można podać wzór (wyrażenie arytmetyczne) na obliczenie wartości, która ma się w tej komórce znaleźć. Komputer ma wykonać obliczenia i sam wpisać ich wynik. A wzór zapamiętać na później, bo jeśli tylko zmienią się występujące we wzorze dane, to musi obliczenie wykonać ponownie i uaktualnić zawartość komórki. Teraz wystarczy tylko dodać, że jako dane we wzorze mogą wystąpić wartości innych komórek, i już jesteśmy w domu. Możemy w naszej tabelce przewidzieć takie obliczenia jakie nam się podoba.

Oczywiście chcemy żeby pisanie wzorów było możliwie najwygodniejsze. Warunek ten będzie spełniony, gdy pozwolimy użytkownikowi umieszczać we wzorach wszystko co tylko może mu się przydać, a więc liczby, typowe funkcje matematyczne (sinus, cosinus, logarytm, pierwiastki itd.), zawartość pojedynczych komórek i całych grup komórek. Np. ważna jest możliwość obliczenia sumy lub iloczynu wszystkich elementów wybranej kolumny lub wiersza. Można też kazać policzyć ile wartości jest wpisanych w zadanym fragmencie tabeli. Typowe funkcje to MAX i MIN — znajdowanie odpowiednio największego i najmniejszego elementu oraz znajdowanie średniej arytmetycznej dla grupy liczb.

Wspólną cechą ostatnio wymienionych działań jest to, że wszystkie operują na grupie komórek. Określić tę grupę można na wiele sposobów — im więcej jest możliwości, tym łatwiej użytkownikowi znaleźć coś dla siebie. Podstawowe to, wspomniana już pojedyńcza kolumna lub wiersz. Dalej, wszystkie komórki z kilku kolumn lub kilku wierszy. I wreszcie, wszystkie dane z prostokąta powstałego na przecięciu grupy kolumn z grupą wierszy.

W tej chwili potrafimy zbudować tabelę pasującą idealnie do naszych potrzeb. Ale gdy już będzie gotowa, to trzeba wprowadzić do niej dane. Nic prostszego, kursor wskazuje na miejsce, w którym program umieści to, co wpiszemy. Używając klawiszy sterujących kursorem możemy go łatwo przesunąć do dowolnej innej komórki. Jako dodatkowe udogodnienie możemy raz wpisane wartości kopiować do innych pozycji tabelki. Przypomnę tu raz jeszcze, że jeśli wstawimy nową wartość w miejscu, które jest użyte we wzorze na obliczanie zawartości innej komórki, to program natychmiast to zauważa, oblicza nową wartość i wpisuje na miejsce starej. Dzięki temu, jeśli np. określiliśmy ostatnią komórkę w kolumnie jako sumę pozostałych wartości, to zmieniając liczby wpisane w tej kolumnie nie musimy już więcej martwić się o sumę — program sam będzie ją na bieżąco aktualizował. Przy okazji można wskazać jeszcze jedno zastosowanie „elektronicznych tabelek”: planowanie, czyli co będzie jeśli...? Przygotowujemy wzory potrzebnych obliczeń, wprowadzamy komplet danych, a potem możemy zmieniać pojedyncze wielkości i obserwować jak zmieniają się wyniki.

Ale wróćmy do wprowadzania danych. Świat jest duży a ekran mały. Jak, w czytelny sposób, zmieścić na małym ekranie wielki fragment? To się niestety nie da zrobić. Ale można inaczej. Cały formularz (inna nazwa naszej tabelki) siedzi w pamięci operacyjnej, a ekran traktujemy jako okienko, przez które można patrzeć na jego fragment. Oczywiście to „okienko” można przesuwać po całym formularzu. Np. jeśli kursor znajduje się w skrajnie prawej kolumnie ekranu i spróbujemy go przesunąć dalej w prawo, to, aby przejść do następnej kolumny, Kursor „popchnie” w prawo cały ekran — pierwsza z lewej kolumna zniknie, a za to pojawi się na ekranie niewidoczna dotychczas kolumna z prawej strony. Oczywiście to samo można zrobić w kierunku góra-dół. Można też, przy pomocy specjalnych komend „skakać” ekranem dalej niż o jedną kolumnę czy wiersz.

Jak zapewne zauważyli stali Czytelnicy, „Przed ekranem” nie publikuje pozycji fantastyczno-nau-kowych. Z tej obserwacji łatwo wysnuć słuszny wniosek, że programy do tworzenia formularzy (ang. „spreadscheet”) istnieją na prawdę a nie w sferze marzeń czy planów. Przykłady — proszę bardzo. Oto nazwy programów, z którymi zetknąłem się na naszym rynku: „**Multiplan**” na IBM PC, „**EasyCalc**” i „**EasyCalc Result**” na Commodore C-64, „**VU-Calc**” na Spectrum.

Andrzej Pilaszek

TAPE COPIER • TAPE COPIER • TAPE COPIER

Jeżeli wykażesz odrobinę cierpliwości to już za godzinę będziesz posiadaczem programu kopiującego, który wzbogaci bibliotekę programów Twojego ATARI 800 XL.

Oto podstawowe własności programu „46K TAPE COPIER":

— kopiuje wszystkie programy taśmowe o standardowej budowie tzn. składające się z bloków rozpoczynających się 20 sek. sygnałem synchronizującym, po nim z serii rekordów zawierających 128 bajtów danych. Program akceptuje zapisy z długimi lub krótkimi przerwami między rekordami, wyprowadza jedynie przerwy krótkie.

Blok rekordów powinien kończyć się rekordem „eof" (end of file) składającym się ze 128 zer,

— program jest bardzo krótki — ma długość 600 bajtów. Czas ładowania do pamięci wynosi 30 sek. Program kopiujący należy ładować podobnie jak większość gier „przez" start/option,

— program potrafi skopiować blok długości ponad 46K bajtów dokładnie 369 rekordów 128 bajtowych. Przeliczając to na czas ładowania otrzymujemy blok ładujący się blisko 16 minut przy standardowej prędkości transmisji 600 bitów/ sek.,

— odczyt lub zapis kopiowanych programów można prowadzić przy wyłączonym głośniku. Program posiada diagnostykę błędów — informuje o niepoprawnym odczycie taśmy źródłowej oraz o poprawnym zakończeniu odczytu lub zapisu,

— pozwala odsłuchać taśmę i dokładnie ustawić początek programu, wgrać przerwę międzyprogramo-wą żądanej długości. Sygnalizuje także zbliżanie się do granicy pojemności wolnej pamięci operacyjnej — wyświetlając ostatnie jeszcze mieszczące się rekordy w górnej części ekranu. Wciśnięcie klawisza „break" w czasie zapisu sygnału synchronizującego powoduje rozpoczęcie zapisu danych oraz eliminacje końcowego rekordu „end to file",

— program odwołuje się do procedur systemu operacyjnego za pośrednictwem tablicy skoków: tak więc powinien dać się uruchomić na innych mikrokomputerach: ATARI 800 oraz 130 XE.

Program został pierwotnie napisany w assemblerze 6502 i skompilowany kompilatorem LADS. W celu udostępnienia go również tym, którzy nie dysponują assemblerem został przetłumaczony na ciąg instrukcji DATA i jest dostępny w BASIC-u.

Zamieszczony obok program składa się z trzech części. Część pierwsza obejmująca instrukcje o etykietach 10-65 jest ładowaczem programu kopiującego i procedury zapisu na taśmie. Instrukcje wprowadzają do pamięci kolejne liczby z bloku data, sprawdzają sumy kontrolne i w przypadku błędu wskazują źle przepisaną instrukcję. Ten zabieg pozwala sprawnie przepisać cały program z dość dużym prawdopodobieństwem, że nie popełni się błędu

Druga i trzecia część programu obejmuje instrukcje DATA. Wiersze 101-110 zawierają przetłumaczoną na ciąg liczb procedurę zapisu programu bootstrapowego (ładowanego przez operację pulpitową „start/option") na taśmę. Procedura posłuży do wyskładowania programu kopiującego. Wiersze 111 — do końca zawierają program kopiujący. Stawiającym pierwsze kroki w programowaniu w assemblerze proponuję przetłumaczenie tych wierszy na assembler i analizę tego programu. Zwracam uwagę na to, że 6 początkowych bajtów programu bootstrapowego ma charakter organizacyjny i nie zawiera rozkazów. Pierwszy wykonywany rozkaz rozpoczyna się w 7 bajcie.

A teraz lista czynności, które należy wykonać aby otrzymać program kopiujący:

1. przepisz uważnie zamieszczony obok program,
2. na wszelki wypadek zapisz go na taśmie,
3. przygotuj kasetę przeznaczoną na program kopiujący,
4. uruchom przepisany program (RUN),
5. jeżeli przepisałeś poprawnie, to po ok. 15 sekundach usłyszysz podwójny sygnał — włącz zapis,
6. jeżeli popełniłeś błąd w czasie przepisywania — program poinformuje Cię o tym. Popraw błędnie przepisaną instrukcję i przejdź do p.4,
7. program, który otrzymałeś załaduj do pamięci podobnie jak ładuje się większość gier i programów użytkowych, naciskając start/option i włączając komputer.

I wreszcie na koniec krótka instrukcja obsługi programu:

— wczytywanie pliku do kopiowania — naciśnij „0". Prawidłowo przeprowadzone wczytanie program sygnalizuje komunikatem „KONIEC"; błąd w czasie odczytu — komunikatem ‚BŁAD",

— zapis pliku — naciśnij „Z". Taśma wyjściowa musi być wcześniej przygotowana, a magnetofon ustawiony na zapis. Zakończenie operacji program sygnalizuje ko-munikatem „KONIEC". Istnieje możliwość skrócenia długości sygnału synchronizującego z jednoczesną eliminacją końcowego rekordu „eof" — w czasie nagrywania sygnału synchronizującego naciśnij „break",

— nagranie przerwy lub odsłuch. Naciśnięcie klawisza „P" pozwala włączyć silnik magnetofonu na okres 10 sekund. Odliczanie sekund można obserwować na ekranie monitora. Przesuw taśmy można zatrzymać w dowolnym momencie naciskaiąc klawisz spacji.

Dariusz Adamowski

```
10 REM LADOWACZ PROGRAMU KOPIUJACEGO
15 N=100:GOSUB 55
20 FOR I=32768 TO 33447
25 READ A:POKE I,A:CS=CS+A:L=L+1
30 IF L=9 THEN GOSUB 45
35 NEXT I
40 X=USR(32768):END
45 READ CSD
50 IF CSD<>CS THEN GOSUB 60
55 CS=0:L=1:N=N+1:RETURN
60 ? "BLAD W WIERSZU ";N;" LUB ";N-1
65 LIST N-1,N:STOP
99 REM PROGRAM KOPIUJACY
100 REM DARIUSZ ADAMOSKI 07,1986
101 DATA 162,16,169,3,157,66,3,169,745
102 DATA 8,157,74,3,169,128,157,75,771
103 DATA 3,169,72,157,68,3,169,128,769
104 DATA 157,69,3,32,86,228,48,36,659
105 DATA 169,11,157,66,3,169,80,157,812
106 DATA 68,3,169,128,157,69,3,169,766
107 DATA 128,157,72,3,169,2,157,73,761
108 DATA 3,32,86,228,48,8,169,12,586
109 DATA 157,66,3,32,86,228,104,96,772
110 DATA 67,58,158,0,0,0,0,0,283
111 DATA 0,5,168,189,174,189,169,60,954
112 DATA 141,2,211,169,128,141,198,2,992
113 DATA 141,200,2,169,14,141,197,2,866
114 DATA 141,196,2,169,51,141,48,2,750
115 DATA 169,191,141,49,2,173,252,2,979
116 DATA 201,8,240,11,201,23,240,37,961
117 DATA 201,10,240,71,24,144,238,32,960
118 DATA 20,191,169,4,32,44,190,162,812
119 DATA 0,189,226,191,157,169,191,232,1355
120 DATA 224,7,208,245,32,107,190,32,1045
121 DATA 73,190,76,205,189,32,20,191,976
122 DATA 169,8,32,44,190,162,0,189,794
123 DATA 233,191,157,169,191,232,224,7,1404
124 DATA 208,245,32,143,190,32,96,190,1136
125 DATA 172,253,191,48,176,32,32,191,1095
126 DATA 76,205,189,32,20,191,32,183,928
127 DATA 190,76,205,189,162,32,157,74,1085
128 DATA 3,169,3,157,66,3,169,128,698
129 DATA 157,75,3,169,223,157,68,3,855
130 DATA 169,191,157,69,3,32,86,228,935
131 DATA 96,162,32,169,12,157,66,3,697
132 DATA 32,86,228,162,32,189,72,3,804
133 DATA 133,212,189,73,3,133,213,96,1052
134 DATA 162,32,169,12,157,66,3,32,633
135 DATA 86,228,96,169,7,162,32,157,937
136 DATA 66,3,169,128,157,68,3,169,763
137 DATA 4,157,69,3,169,255,141,72,870
138 DATA 3,157,73,3,32,86,228,192,774
139 DATA 136,208,40,32,32,191,96,169,904
140 DATA 11,162,32,157,66,3,169,128,728
141 DATA 157,68,3,169,4,157,69,3,630
142 DATA 165,212,157,72,3,165,213,157,1144
143 DATA 73,3,32,86,228,140,253,191,1006
144 DATA 48,1,96,32,1,191,96,169,634
145 DATA 52,141,2,211,169,15,133,140,863
146 DATA 169,10,133,139,230,140,165,140,1126
147 DATA 141,172,191,173,252,2,201,33,1165
148 DATA 240,37,169,6,141,0,191,162,946
149 DATA 137,160,255,136,192,0,208,251,1339
150 DATA 202,224,0,208,244,206,0,191,1275
151 DATA 169,0,205,0,191,208,232,198,1203
152 DATA 139,169,0,197,139,208,205,169,1226
153 DATA 60,141,2,211,32,32,191,96,765
154 DATA 0,162,0,189,240,191,157,169,1108
155 DATA 191,232,224,7,208,245,169,255,1531
156 DATA 141,252,2,96,162,0,138,157,948
157 DATA 169,191,232,224,7,208,248,96,1375
158 DATA 162,0,189,246,191,157,169,191,1305
159 DATA 232,224,7,208,245,169,255,141,1481
160 DATA 252,2,96,112,112,112,66,40,792
161 DATA 189,2,2,112,70,83,191,112,761
162 DATA 112,112,6,112,112,112,112,112,790
163 DATA 112,2,112,6,112,112,2,65,523
164 DATA 51,191,1,0,0,0,0,97,340
165 DATA 116,97,114,105,0,24,16,16,488
166 DATA 0,120,108,0,0,0,0,0,228
167 DATA 0,20,22,107,0,0,116,97,362
168 DATA 112,101,0,99,111,112,105,101,741
169 DATA 114,0,0,0,0,0,0,0,114
170 DATA 0,175,100,99,122,121,116,0,733
171 DATA 0,0,0,0,186,97,112,105,500
172 DATA 115,0,0,0,0,0,176,114,405
173 DATA 122,101,114,119,97,0,0,0,553
174 DATA 0,0,0,0,0,0,0,0,0
175 DATA 0,0,0,0,0,0,0,0,0
176 DATA 0,0,0,0,0,0,0,36,36
177 DATA 97,114,105,117,115,122,0,33,703
178 DATA 100,97,109,111,119,115,107,105,863
179 DATA 0,100,108,97,0,99,122,121,647
180 DATA 116,101,108,110,105,107,111,119,877
181 DATA 0,34,33,42,52,43,33,67,304
182 DATA 58,155,111,100,99,122,121,116,882
183 DATA 0,0,122,97,112,105,115,0,551
184 DATA 0,98,108,97,100,0,0,107,510
185 DATA 111,110,105,101,99,0,0,0,526
```

ŚWIAT DŹWIĘKÓW ATARI cz. I

Atari posiada duże możliwości dźwiękowe. Pozwalają one wykorzystać ten mikrokomputer jako instrument muzyczny, a nawet jako syntezator mowy ludzkiej (np. program S.A.M.). Wiadomo, że dochodzenie do perfekcji w grze na jakimś instrumencie wymaga długiego czasu. Także opanowanie możliwości muzycznych jakie daie Atari wy-maga czasu i znajomości komputera. Spróbujmy zapoznać się z niektórymi możliwościami, jakie daje Atari.

W BASIC-u mamy do dyspozycji instrukcję **SOUND**. Posiada ona cztery parametry i jej działanie trwa tak długo, aż nie zostanie wykonana następna instrukcja **SOUND** (dla tego samego generatora), **END**, **RUN**, lub **NEW**. Postać tej instrukcji jest następująca:

SOUND c, p, d, v

gdzie:

c — numer generatora (od 0 do 3)

p — okres generowanego dźwięku (od 0 do 255)

d — określa brzmienie (liczby parzyste od 0 do 14)

v — jest głośnością (od 0 do 15)

Atari ma cztery niezależne generatory dźwięku. Można jednocześnie uruchomić wszystkie generatory, lub tylko kilka z nich.

Okres generowanego dźwięku określa jego wysokość. Jak wiemy okres ten jest odwrotnie proporcjonalny od częstotliwości. Warto znać dokładną zależność pomiędzy tymi dwoma wartościami. Jest ona następująca:

częstotliwość=31960/(okres+1) [Hz]

Okazuje się jednak, że możliwości dźwiękowe Atari są znacznie większe, niż dawane przez instrukcję **SOUND** w języku BASIC. Aby je wykorzystać musimy poznać bliżej sposób generowania dźwięku w Atari.

Generowaniem dźwięku zajmuje się specjalizowany układ scalony nazwany **POKEY**. Posiada on własne rejestry zawierające informacje o wytwarzanych dźwiękach. Wyróżnić możemy trzy obszary sterowania dźwiękiem. Nazwijmy je tak, jak stosuje to w swoich publikacjach sama firma Atari: **AUDF, AUDC i AUDCTL**

AUDF 1 do 4. Zlokalizowane są w komórkach o adresach **53760, 53762, 53764, 53766**. Określają one wysokość generowanego tonu. Na przykład instrukcja **POKE 53762,64** odpowiada instrukcji **SOUND 1,64,x,x** (wartość x jest tutaj nieistotna).

AUDC 1 do 4. Zlokalizowane w komórkach o adresach **53761, 53763, 53765, 53767**. Ich znaczenie jest bardziej skomplikowane niż **AUDF**. Przyjrzyjmy się im dokładniej. Cztery najmniej znaczące bity każdej z tych komórek sterują głośnością odpowiedniego generatora. Cztery bity czyli w systemie dziesiętnym liczby z zakresu od 0 do 15, jest to tyle, ile stopni głośności można zapisać w instrukcji **SOUND**. Piąty bit jest to tzw. bit dwustanowej regulacji głośności. Pozostałe trzy bity określają zakłócenia. Jest to szerokie pole do eksperymentowania. Na trzech bitach można zakodować liczby dziesiętne od 0 do 7. Jak pamiętamy w instrukcji **SOUND** parametrem określającym zniekształcenia może być liczba parzysta z zakresu od 0 do 14. Takich liczb jest właśnie osiem. Jeżeli wszystkie bity posiadają wartość jedynki logicznej to otrzymujemy czysty dźwięk. Odpowiada to umieszczeniu w instrukcji **SOUND** parametru sterującego brzmieniem równego 14.

Zilustrujmy to przykładem: Instrukcji **SOUND 1,64,10,10** odpowiada **POKE 53762,64: POKE 64363,234. AUDCTL** jest to komórka o adresie **53768**. Zawartość tej komórki zmieniać możemy tylko instrukcją **POKE**. Wykonanie instrukcji **SOUND** nie powoduje zmiany zawartości tej komórki. Wykorzystując odpowiednio tę komórkę możemy uzyskiwać efekty niemożliwe do osiągnięcia tylko przez instrukcję **SOUND**. Znaczenie poszczególnych bitów **AUDCTL** jest następujące:

BIT 0 przełącza główny zegar bazowy z 64 kHz na 15 kHz

BIT 1 włącza filtr górnoprzepustowy dla kanału 2 sterowany kanałem 1

BIT 2 włącza filtr górnoprzepustowy dla kanału 1 sterowany kanałem 3

BIT 3 powoduje połączenie kanałów 3 i 4 pozwalając używać 16-bitowej zmiennej sterującej

BIT 4 powoduje połączenie kanałów 1 i 2 pozwalając używać 16-bitowej zmiennej sterującej

BIT 5 powoduje przełączenie zegarów kanału 3 na 1.79 MHz

BIT 6 powoduje przełączenie zegarów kanału 1 na 1.79 MHz

BIT 7 przełącza licznik „poly” z 17 na 9 bitów

Znaczenie tych bitów wymaga dokładniejszego wyjaśnienia.

BIT 0. Nie zagłębiając się teraz w szczegóły można powiedzieć, że im wyższa jest częstotliwość zegara, tym wyższe dźwięki można uzyskać. Zegar bazowy ma normalnie częstotliwość 64 kHz. Przełączając go na częstotliwość 15 kHz można generować dźwięk o niższej częstotliwości.

BIT 1 i 2. Filtry generatora dźwięku Atari nie są filtrami analogowymi. Ich działanie polega na zmianie stopnia wypełnienia impulsów, dzięki czemu generowanych jest więcej harmonicznych o wysokich częstotliwościach. W uproszczony sposób można powiedzieć, że taki filtr nie przepuszcza częstotliwości mniejszych, niż częstotliwość kanału sterującego.

BIT 3 i 4. Konieczność łączenia dwóch generatorów występuje gdy chcemy uzyskać szerszy przedział generowanych dźwięków, lub gdy konieczne jest uzyskanie płynnej zmiany częstotliwości w zakresie częstotliwości wysokich. Łącząc dwa kanały razem uzyskujemy 16-bitową zmienną sterującą dzięki czemu zmniejszamy najmniejszy skok częstotliwości z 1/256 (jeden bajt) do 1/65536 (dwa bajty), ilustruje to poniższy przykład:

```
10 SOUND 0,0,0,0
20 FOR X=55 TO 0 STEP -1
30 SOUND 0,X,10,10
40 FOR F=1 TO 55:NEXT F
50 NEXT X
60 SOUND 0,0,0,0
```

Uruchamiając ten program usłyszymy wyraźnie, że przy wysokich dźwiękach częstotliwość wzrasta skokowo. Teraz napiszmy program wykorzystujący AUDCTL dla połączenia dwóch kanałów razem i uzyskania płynnej zmiany.

```
10 SOUND 0,0,0,0
15 POKE 53768,80
20 POKE 53761,160
25 POKE 53763,234
30 FOR X=0 TO 255
35 POKE 53762,X
40 FOR Z=0 TO 255
45 POKE 53760,Z
50 NEXT Z
55 NEXT  X
60 SOUND 0,0,0,0
```

W powyższym programie wykorzystaliśmy większość zdobytych dotąd wiadomości. Poszczególne jego linie realizują następujące zadania:

Linia 10 inicjuje **POKEY**.

Linia 15 ustawia w **AUDCTL** bit 2, 4 i 6.

Linia 20 nadaje wartość **AUDC 1** i wyłącza pierwszy generator dźwiękowy.

Linia 25 ustawia **AUDC 2**, włącza generator drugi, włącza czyste tony oraz ustawia głośność 10.

Linie 30 do 55 dwie pętle **FOR...NEXT** służą do nadawania wartości zmiennym **AUDF 1** i **AUDF 2**. Wartością zmiennej sterującej regulujemy wartość częstotliwości zgrubnie, natomiast pierwszym generatorem dokładnie.

Powróćmy teraz do opisu pozostałych bitów **AUDCTL**.

BITY 5 i 6. Działanie ich jest zbliżone do działania bitu O z tą różnicą, że zwiększają one częstotliwość generowanego tonu. Bit 5 zwiększa częstotliwość zegara do 1.79 MHz w kanale 3, natomiast bit 6 w kanale 1. Wykonanie poniższych instrukcji zobrazuje działanie tycn bitów.

SOUND 0,255,10,10
POKE 53758,64

BIT 7. Ustawia pojemność licznika używanego przez generator szumów. Im większa pojemność tego licznika, tym częściej powtarzane są zakłócenia generowanego dźwięku. Spróbujmy wykonać następujące instrukcje:

SOUND 0,90,8,10
POKE 53769,128

Mariusz J. Giergiel



ENIGMA FORCE



SEVERINA

wylęgarnia insektów

zbrojownia insektoidów

SYYLK

MAUL

ZARK


KOLOROWE KLUCZE


KLUCZ ZOFFA


GRANAT


NARZĘDZIA SEVERINY


RYBA


LASER


PARALIZER





RT

KONIEC

Bajtek

ENIGMA FORCE

ENIGMA — bojowa organizacja w służbie Imperatora. Należy do niej elita niezwyciężonych legionów Imperium. Cechuje ich ogromna waleczność. Walczą z determinacją i poświęceniem, za Imperatora oddadzą życie. Najnowsze uzbrojenie i nowoczesne wyszkolenie techniczne grupy podnosi ich walory bojowe. Członkowie organizacji po przeszkoleniu zostali posłani do kryjówki gen. Zoffa, imperialnego zdrajcy i wyjątkowego łajdaka, z misją SHADOW FIRE. Po pierwszej bitwie stoczonej z Zoffem, Enigmę opuścili Torik i Manto, „leczą" swe rany na planecie Styx. Pozostała czwórka została wysłana na dalsze poszukiwania. Zoffa dopadli na planecie Hetera w jego tajnej bazie i tam rozstrzygną się losy zdrajcy.

OSOBY:

ZARK MANTER

POCHODZENIE: człowiek
WIEK: 39 lat (wiek fizyczny)
STANOWISKO: dowódca grupy
DANE OSOBOWE: Rekrut legionów Imperatora. Podczas bitew odniósł wiele ran, jest poczęści cyborgizowany, sztuczne prawe przedramię.
UWAGI OGÓLNE: Duża wiedza o różnych typach broni, ekspert od walki grupowej. Tłumacz.

SYYLK

POCHODZENIE: insektoid
WIEK: brak danych
STANOWISKO: z-ca dowódcy grupy.
DANE OSOBOWE: Rekrut legionów. Bezlitosny i odważny. Patologiczna nienawiść do Zoffa za podbój jego ojczyzny. Małomówny, samotnik.
UWAGI OGÓLNE: Dobry żołnierz. Nie rozstaje się nigdy z bronią. Bardzo inteligentny. Gra w szachy. Silny psychicznie.

SEVERINA MARIS

POCHODZENIE: człowiek (kobieta)

WIEK: 23 lata (wiek fizyczny)

STANOWISKO: specjalista-technik

DANE OSOBOWE: Uparta. Rekrut z celi śmierci na Kerolth. Dzięk rekomendacji TOMKA przyjęta do ENIGMY. Groźna, niezależ na, lojalna tylko wobec sie bie, rutynowany technik.

UWAGI OGÓLNE: Doskonały strze lec i ślusarz. Nie lubi białe go koloru.

MAUL

POCHODZENIE: robot bojowy
WIEK: data opuszczenia taśmy prod. 30.07.2112.
STANOWISKO: robot bojowo-obronny
DANE OSOBOWE: —
UWAGI OGÓLNE: Ciężar 185 kg, wys. 126 cm, napęd antygrawitacyjny. Porusza się dosyć wolno, lecz posiada laser dużej mocy przydatny w walce.

Cel misji w bazie Zoffa — zabić generała Zoffa i zniszczyć pamięć komputera bazy.

Spis poszczególnych rzeczy:
KEYCARD — (karta-klucz) karta perforowana z zapisanym na niej (cyfrowo) kodem otwierającym konkretne drzwi. Każdy kolor karty do innych drzwi.
ZOFFCARD — (karta Zoffa) karta z kodem znaleziona przy zwłokach Zoffa. Otwiera drzwi oznaczone END.
RYBA — symbolizuje dawkę energii dla potrzebującej postaci. Energia, po użyciu ryby, wzrasta na max.
NARZĘDZIA — zbiór przyrządów w obsłudze znanych tylko!!! Severinie. Ona może nimi otworzyć każde (prócz tych oznaczonych przez END) drzwi bez większego trudu. Uwaga! nikt nie może jej przy tym przeszkadzać.
GRANAT — sposób użycia i przeznaczenie znane.
LASER — broń dużej mocy, „mocniejszy" od PARALIZERA.
PARALIZER — broń średniej mocy, dobry na małe odległości, nietrwały.

W bazie Zoffa nigdy nie ma zgody wśród podwładnych a część z nich tzn. insektoidzi przeszli na stronę IMPERIUM, trzeba to wykorzystać. Insektoidzi wyglądem przypominają SYYLKA lecz mają ciemniejszą powłokę. Tylko SYYLK może przekonać Insektoidów do podjęcia decyzji.

Najkrótsza droga do celu:
1. Podziel znalezione przedmioty pomiędzy członków grupy.
2. Poszukaj karty-klucza idąc w dół a potem w górę a następnie wróć na start.
3. Wyślij (koniecznie SYYLKA) na lewo od miejsca startu potem w górę i jeszcze raz w górę, otwórz drzwi kartą a potem trzy razy w lewo aż do zbrojowni Insektoidów.
4. Po drodze rozglądaj się za dowolnym z Insektoidów który powie ci: „Rozmawiałem z Królową. Jest wierna Imperium i jego Władcy" — po tej rozmowie żaden z Insektoidów już Cię nie zaatakuje. Insektoidzi pomogą Ci w akcji.
5. Sprowadź całą grupę do zbrojowni Insektoidów. Załóż tam swoją „bazę wypadową".
6. Po dozbrojeniu poprowadź całą grupę do pomieszczenia, gdzie znajdziesz narzędzia dla Severiny tj. w lewo, w górę, w prawo, w prawo, w górę. w górę, w prawo, w dół, w górę. Narzędzia od razu daj Severinie.
7. Teraz gdy masz już większość potrzebnych przedmiotów musisz odnaleźć gen. Zoffa. W tym celu wyślij, uzbrojonych w granaty, SYYLKA lub SEVERINE, gdyż oni są najlepszymi strzelcami. Gen. Zoffa można zastrzelić z broni ręcznej lecz wymaga to czasu i dobrego ustawienia się, ale można też wrzucić granat do pomieszczenia, gdzie aktualnie jest: Zoff, po jego detonacji przeszukaj tamten pokój i zabierz ZOFFCARD (kartę Zoffa).
8. Wróć do reszty tj. do pomieszczenia, gdzie były narzędzia i przejdź do zbrojowni Repliansów tzn. trzy razy w prawo, dwa w górę, w prawo, w górę, dwa razy w dół, w prawo i otwórz narzędziami lub inną znalezioną kartą drzwi oznaczone na mapie iako END i zejdź na dół do zbrojowni. Dobrze jest zamknąć za sobą drzwi.
9. Po użyciu ZOFFCARD, karty Zoffa, wyślij całą grupę w dół przez drzwi oznaczone jako END by znaleźć się w sali komputerów i zakończyć misję pełnym sukcesem.

Czynności, które może wykonać każdy członek grupy:
PICK UP — podnoszenie przedmiotów... nastaw krzyż na obrazek potem ENTER (FIRE) i weź przedmiot,
DROP — upuszczenie przedmiotu... jak PICK UP,
ACTIVATE — uruchomienie (użycie) przedmiotu... nastaw krzyżyk na obrazek potem ENTER (FIRE) potem wskaż konkretny przedmiot,
WEAPON — broń... służy do zamiany broni lub aktywacji tejże,
OOPS — kasowanie ostatniej wydanej komendy ENIGMA FORCE.

autor: DENTON DESIGNS
producent: BEYOND Challenging Software
komputer: ZX-Spectrum 48k, Commodore C64/128
ponadto: nagroda za muzykę wersji CBM 64 (Brilant Sound-Track"

(ML)



GLOBTROTER

10 000 marek i 60 dni — czy to wystarczy?

Niewielu szczęśliwcom udaje się zrealizować marzenia o podróżach szlakami Marco Polo, Magellana czy Kolumba. Ktoś zazdrości stawy bohaterom powieści Julesa Verne „W osiemdziesiąt dni dookoła świata”, a nie stać go na realizację ich pomysłu. Inny zmusza swoje dziecko, aby pokazało na mapie, gdzie leży Pernambuco, a latorośl twardo wskazuje na Rzym.

Szkoła nie zapewnia dobrej znajomości geografii świata. Pomocą dla tych, którzy chcieliby orientować się i uchodzić za bywałych w różnych krajach może się okazać program firmy DATA MEDIA ATARI ELEKTRONIK pod nazwą GLOBTROTER.

Gra rozpoczyna się zawarciem niezbyt fortunnego zakładu. Na przyjęciu w obecności wielu osób świadomie stwierdziliśmy, że odwiedzimy co najmniej dziewięć wybranych miast na świecie w czasie krótszym niż 60 dni, dysponując zaledwie kwotą 10 000 marek.

Klamka zapadła, musimy spróbować aby zachować twarz. Podczas podróży będziemy narażeni na różne niedogodności, z jakimi spotykają się turyści. Podstawowym naszym kłopotem może okazać się brak pieniędzy. Ale i na to jest rada. Trzeba będzie je po prostu wygrać. W Zurychu spróbujemy na giełdzie, w Los Angeles w automatach do gry, w Londynie w bingo, w Fremantle na wyścigach konnych, w Hanoi pobawimy się łamigłówką, w Nicei ruletką, a w Miami zagramy w golfa.

Drogę naszej wędrówki wybieramy według własnego uznania. Przed nami cały świat. Zabawę rozpoczynamy i kończymy w Hamburgu.

Podczas podróży możemy korzystać z samolotu, pociągu, statku, oraz wynajętego samochodu pamiętając o tym, ze im szybszy środek transportu, tym drożej za niego płacimy. W zabawie mogą uczestniczyć cztery osoby.

GLOBTROTER bywa złośliwy. Od czasu do czasu oznajmi, że właśnie wydaliśmy kilkadziesiąt marek na Małe Conieco w przydrożnym barze lub kupiliśmy nowe ubranie za parokroć więcej. Zdarzy się, że nas okradną — konsulat zazwyczaj refunduje od 60 do 90% naszych zasobów finansowych, lecz gdy tego nie uczyni — żegnajcie marzenia o dalszej podróży. Czasami jednak lokalna prasa zapłaci nam kilkaset marek za wywiad z obieżyświatem albo kupiony na loterii los okaże się szczęśliwy.

Jak każda gra komputerowa, tak i GLOBTROTER nie jest wolny od niekonsekwencji, a nawet błędów. Do pierwszych zaliczyć trzeba takie paradoksy, jak podróż z Paryża do Londynu najkrótsza trasa prowadzi przez Frankfurt i Amsterdam. Niemałym zaskoczeniem dla znawcy Azji będzie stwierdzenie, że z Singapuru można polecieć co najwyżej do Honkongu — połączeń międzykontynentalnych brak! Powód tych ograniczeń jest jasny — bez nich zbyt łatwo można by wygrać grę; mocno jednak cierpi na tym efekt edukacyjny. Niewątpliwym błędem w zestawie danych programu jest morska podróż z Tangeru (Afryka) do Panamy (Ameryka Środkowa) — tańsza i krótsza niż przejażdżka pociągiem z Rzymu do Neapolu (200 km). Podobny szok przeżywamy w Colombo stwierdzając, że pociąg do Hanoi jedzie tyle czasu, co do Kalkuty.

Natomiast zaletą programu GLOBTROTER jest fakt, że grę możemy przerwać w dowolnym momencie i wrócić do niej za godzinę, dzień, miesiąc czy rok dokładnie w to samo miejsce, w którym się zatrzymaliśmy. W trakcie gry możemy zażądać pokazania mapy poszczególnych kontynentów z wymienionymi głównymi miastami. Zawsze można zrezygnować z wybranego już środka transportu. Na bieżąco informowani jesteśmy o stanie naszego konta i aktualnym miejscu pobytu.

Życzę szczęśliwej podróży! *(spi)*



ZORRO

Dwa miesiące temu dostałem od przyjaciela grę firmy DATASOFT INCORPORATION noszącą, znany wszyst kim trzydziestolatkom i nie tylko, tytuł ZORRO. Bardzo się ucieszyłem z tego prezentu, bo od razu przypomniały mi się lata, kiedy jako dziecko ganiałem z kijem od szczotki. Teraz komputer ATARI daje wszystkim, małym i dużym,możliwość zabawy w meksykańskiego Janosika.

Rozpocząłem grę od zwiedzania miasteczka, wdając się przy tym bezustannie w pojedynki na szpady z wychodzącymi z zaułków i bram gwardzistami, lecz nic z mojego chodzenia nie wynikało. Postanowiłem się poradzić i zatelefonowałem do ośmioletniego dziecka przyjaciela. To był dobry pomysł.

ZORRO został odarty z tajemnic. Kierowany drążkiem sterowym poszedł czym prędzej po czarną chusteczkę rzuconą mu przez wybrankę jego serca z balkonu, zawieszoną przez przypadek na korbie od studni. Potem przeniósł się do innej części miasteczka po klucz do drzwi komnaty, w której zamknięty był żelazny pręt do znakowania bydła. Z prętem popędził do kuźni, gdzie rozgrzał go do czerwoności. Zastanawiałem się po co, ale w chwilę później ZORRO napiętnował nim bezpańskiego byka. Potem odszukał podkowę, którą zabrał ze sobą na szczęście. Na tym etapie gry pojawiły się na ekranie, jeden po drugim, dwa dzwony, które tytułowy bohater powiesił na starej dzwonnicy nieopodal cmentarza. Wówczas otworzył się grobowiec.

ZORRO długo zastanawiał się, czy podkowa jest wystarczającym atrybutem szczęścia, aby wejść do krainy cieni. Postanowił jeszcze raz dokładnie spenetrować miasteczko, co okazało się bardzo rozsądne: po tym spacerze zniknął w fałdach jego płaszcza znaleziony w studni złoty kielich. Kiedy przechodził obok dzwonnicy odegrał sygnałówkę natrąbce i stał się właścicielem buta.

Dopiero teraz skierował swe kroki do grobowca. Obyło się bez niespodzianek. Znalezione tam pieniądze w sakiewkach umożliwiły mu życie długie i szczęśliwe Zastanawia mnie tylko jedno: kiedy już wychodził, pojawiły się sekretne drzwi z namalowanymi na nich podkową, kielichem i butem. Ciekawe co jest za nimi? Myślę, że Wy w roli ZORRO spokojnie poradzicie sobie z tym problemem. *(spi)*



Już siedem miesięcy liczy sobie nasza Bajtka. Na nasze ósme notowanie Bajtkowej Listy Przebojów gier mikrokomputerowych nadeszło 2530 propozycji. Głosowano na 383 gry. Na I miejsce niespodziewanie awansował „BOMB JACK”. Pozostałe miejsca bez wielkich niespodzianek.

1	BOMB JACK	⬆
2	ENIGMA FORCE	⬆
3	MOVIE	⬆
4	FAIRLIGHT	⬇
5	BACK TO SKOOL	⬆
6	RASPUTIN	!
7	SPY VS SPY	!
8	ZORRO	!
9	ELITE	!
10	ROAD RACE	⬇

Wśród wszystkich czytelników, którzy nadesłali nam swoje propozycje do Bajtkowej Listy Przebojów zgodnie z obietnicą rozlosowaliśmy nagrody, programy, które ufundowała brytyjska firma ELECTRONICS EXPORT. Nagrody wylosowali: Rafał Wilk z Świętochłowic, Krystyna Dębska z Józefowa, Janusz Ninard z Radomia i Grzegorz Skórzewski z Warszawy. *Sławek*



SZCZĘŚCIARZ

Piotr Skórzewskl, lat 13, zamieszkały Warszawa, ul. Goławicka 3, uczeń VII klasy Szkoły Podstawowej nr 114 — „szczęściarz” w konkursie Bajtkowej Listy Przebojów.

Twoja propozycja złotej dziesiątki gier została wylosowana i nagrodzona programem „ONE MAN AND HIS DROID” f-my MASTER TRONIC.

— Jaki masz mikrokomputer?

— Niestety nie mam żadnego..., jeszcze. Rodzice obiecują ale jest to dosyć droga zabawka.

— Jak w takim razie typowałeś propozycję na Listę?

— Mój kolega ma ATARI 800 XL. U niego mogę czasem bawić się grami.

— Co zrobisz z nagrodą?

— Poczekam, aż będę miał mikrokomputer.

SAM

Od dłuższego czasu wśród użytkowników C-64 krąży program syntezy mowy „SAM”. Program ten dodaje do zbioru komend BASIC-a kilka własnych rozkazów, pozwalających na uzyskanie całkiem poprawnie brzmiącej mowy. Niestety, bardzo niewielu osobom udało się zdobyć informacje o działaniu tego programu, który przez to bardzo rzadko jest wykorzystywany. Poniżej podaje spis komend oraz sposób ich użycia.

SAY — powiedz — jest to odpowiednik rozkazu PRINT, pozwalający na wymawianie słów i zdań.

]RE — reciter— ustalenie rodzaju pracy polegającego na wymawianiu normalnie wpisywanych słów zgodnie z zasadami angielskiej fonetyki.

]SA — sam — jest to rodzaj pracy, w którym teksty muszą być odpowiednio zapisane (złożone z fonemów), ale za to dużo większe są możliwości wymowy wyrazów.

]PLxx — pitch — wysokość dźwięku. Tu i w następnych linijkach xx oznacza parametr w zakresie 0 — 255, złożony z cyfr (nie można używać zmiennych).

]SPxx — speed— szybkość wymowy

]KNxx,xx — throat — mouth, czyli usta — gardło, ustalenie brzmienia głosu. Normalne parametry to 128, 128.

]ER — error — obsługa błędu. Jeśli przy pracy w trybie „sam” użyjemy niewłaściwych znaków, rozlegnie się akustyczny sygnał błędu, a **]ER** spowoduje pokazanie błędu.

Wszystkie komendy zaczynające się od znaku | interpretowane są podobnie jak **GOTO**, tzn. parametr może składać się tylko z konkretnych cyfr. Jeśli jednak ktoś chce zmieniać parametry mowy za pomocą zmiennych, można to zrobić w następujący sposób:

pitch: **POKE 39439, x**
speed: **POKE 39438, x**
throat: **POKE 38881, x : SYS 38882**
mouth: **POKE 38880, x : SYS 38882**

W niektórych przypadkach komendy SAMa trzeba będzie oddzielić dwukropkiem, np.

IF A = 1 THEN :]PI15 : SAY” ...

Brak dwukropka po **THEN** w powyższym przykładzie spowodował by przerwanie wykonywania programu i dobrze nam znany **SYNTAX ERROR**, czyli błąd składni.

Na zakończenie kilka słów o pisaniu fonetycznie. Jeśli chcemy, aby komputer zrozumiale wymawiał polskie słowa, musimy odpowiednio je zapisać. Ogólnie rzecz biorąc, zasada jest prosta. Po każdej samogłosce wstawiamy literę **H** i cyfrę z zakresu 1 — 8, która decyduje o intonacji w danym miejscu słowa. I tak np. wyraz „BAJTEK” zapiszemy w następujący sposób: **BAH4IH7TEH8K**. Cyfry są oczywiście zależne tylko od własnego uznania. Dla wymowy znaczenie mają też takie znaki jak kropka, przecinek, dwukropek itp. Przy pisaniu fonemami np. znak * (gwiazdka) zastępuje dźwięk „sz” lub „ś”. Ta sama gwiazdka użyta w drugim rodzaju pracy (SHE) zostanie wymówiona jako „asterisk”

Zamieszczony program pokazuje sposób wykorzystania programu **SAM**. Uwaga dla tych, którzy **SAM** będą wczytywać z taśmy — po skończonym ładowaniu zamiast **RUN** należy wykonać **SYS 64738**, a potem **SYS 2064**. Taki sposób uruchomienia gwarantuje poprawną pracę programu. Na pytanie programu „low or high memory” należy odpowiedzieć, naciskając literę **H**. **SAM** wyklucza obecność programów takich jak **TURBO, FAST LOAD, SIMON'S BASIC** itp.

(M.S.)

```
SAM DEMO

10 REM SAM DEMO
20 SAY"MOHVIH SAHM, PROHGRAHM SYNTEHZY MOHVY."
30 SAY"OHTOH P*YKUHAHD VYKOH*YSTAHNEHAH FOHNEHMUHV."
40 SAY"AH7 KUH3 KUH6."
50 SAY"DOH8 REH7 MIH6 FAH5 SOH4 LAH3 SIH2 DOH1 OH2O H1OH 2OH1"
60 SAY"TAH1KIH1 GUH1OH1S IH1EH1ST TEH1* T*IH1EH1KAH1VYYY."
70 SAY"AH7LEH7 TAA4KIH4 TEH6* MOH7*EH7 BY5T*."
80 SAY"POH4LSKAH6 VYMOHVAH3 IHEH7ST TSAH4UHKIHEH6M POH6PRAH8VNAH."
90 SAY"NAH4 P*Y5KUH6AH7DTT"
100 SAY"/H*OHN*T* B*MIH7 V T*T*IH5NIHEH V *T*EHB*EH*Y6NIH6EH."
110 SAY"AH7 TEHRAHZ VYMOHVAH AHNGIHEHLSKAH." 120 ]RE
130 SAV"I LOVE YOU."
140 SAY"BYTECK  IS THE BEST COMPUTER MAGAZINE IN THE WORLD."
150 SAY"AND COMMODORE 64 IS THE BEST COMPUTER."
160 SAY"DO YOU UNDERSTAND ME?"
178 SAY"AE TARAEZ TROHE        AEFEKTOOV."
188 ]SP200:]PI200:SAY"I AM IRON MAN"
198 ]KN100,150:]SP50:]PI100:SAY"MAM GUPY GUUOS"
280 ]SP40:]PI30:]KN40,40 SAY"MAM YESHTCHE GUPSHY GUUOS."
210 ]PI80:]SP60:]KN128,123:SAY"TROHAE MOOZYKEE."
220 A$="EREREREERRTRTTYTYTBTBTHTGT"
230 B$="TATARARERERYFUFYFYFYY"
240 ]KN20,30
250 FOR A=1 TO 4:SAY A$:NEXT
260 FOR A=1 TO 4:SAY B$:NEXT
270 ]KN220,30:]SP20
280 FOR A=1 TO 4: SAY A$: NEXT
290 FOR A=1 TO 4: SAY B$: NEXT
300 GOTO 280

READY.
```



Uff!

Po tak solidnej dawce teorii warto by było trochę odpocząć i przejść do nieco innej tematyki — jakie programy nadają sie do przepisania na taśmę.

Generalną zasadą powinno być uprzednie dobre poznanie danego programu zanim przystąpi się do prób jego przeniesienia na kasetę. Należy przy tym zwrócić uwagę na następujące szczegóły:

— czy komputer W TRAKCIE działania programu kontaktuje się ze stacją dysków?

— czy komputer PO ZAKOŃCZENIU nie zapisuje wyników w jakimś zbiorze na dysku?

W pierwszym wypadku; dla wszystkich nie znających języka maszynowego mamy propozycję — przekonajcie sami siebie, że tak naprawdę to wcale Wam za bardzo nie zależy na tym programie. Taki program będzie wyma-



Pewnego dnia, mając chwilę wolnego czasu, postanowiłem nieco odświeżyć swój zbiór programów. Udałem się w tym celu do bogatego w nowości znajomego, z którym od dawna wymieniam programy. Mieliśmy już przystąpić do przegrywania, gdy nagle pojawiła się trudna do pokonania przeszkoda. Otóż ów znajomy stal się zagorzałym zwolennikiem **TURBO SAVE-LOAD**. Prawie wszystkie jego kasety były nagrane przy użyciu tego programu.

System **TSL** jest bardzo wygodny dla użytkowników **DATASETTE**, daje on możliwość szybkiego wczytywania programów bez konieczności osobnego wgrywania

gał na pewno sporych przeróbek, co jest możliwe ale dla bardziej wtajemniczonych. W drugim wypadku należy po prostu wyłączyć/stację dysków z sieci przed zapisaniem wyników; jeżeli po próbie ich zapisania komputer nie zablokował się i gra wróciła do początku — taki program na taśmie da się zapisać.

W większości wypadków programy jednoczęściowe dają się przenieść bez większych problemów, oczywiście pod warunkiem poprawnego wykonania tej operacji. Dotyczy to wszystkich programów zaczynających się od adresu 2048 i uruchamianych za pomocą RUN. Z naszych obserwacji wynika, że bardzo ważny jest program przyspieszający.

Jak już wiesz, istnieją także programy, które należy wczytywać za pomocą LOAD „NAZWA", 8, 1. W takim wypadku należy zawsze sprawdzić najpierw adres pamięci od jakiego program zaczyna się wczytywać. W tym celu trzeba posłużyć się podstępem i potraktować nasz program jako zbiór; sztuczka polega na odczytaniu dwóch pierwszych bajtów zawierających adres wczytywania.

Tytuł programu musi być wprowadzony dokładnie tak samo jak na dysku, w przeciwnym wypadku otrzymasz bzdury. Mamy już więc nasz adres i ... jednocześnie następny problem. Przypuśćmy, że otrzymałeś adres 680. W tej sytuacji Czytelników bez przygotowania w zakresie języka maszynowego odsyłamy do przedstawionego już rozwiązania — zrezygnowania z prób przegrywania. Takie programy bowiem zmieniają najczęściej wektory ważnych procedur systemu operacyjnego i wymagają bardzo dokładnej analizy pomijając już fakt samouruchamiania; jeżeli ktoś bardzo będzie chciał „rozgryzać" taki program np. poprzez likwidację samouruchamiania to musi robić to bezpośrednio na dyskietce, a nie w pamięci komputera. Inna metoda to zastosowanie dobrego monitora języka maszynowego. Uprzedzamy w każdym razie, że nad takimi programami trzeba się nieźle nabiedzić, a najbardziej potrzebna będzie tu wiedza o komputerze i jego systemie operacyjnym.

Przejdźmy teraz do programów kilkuczęściowych — dużo takich programów bowiem można spotkać na dyskietkach i niektóre z nich są dość łatwe do przeniesienia na taśmę, co wcale nie znaczy, że będzie to proste! Wyobraźmy sobie, że chcemy przenieść na taśmę dwuczęściowy program składający się z krótkiego programu wczytującego oraz właściwego programu:

10 LOAD „ABCD",8,1 20 SYS 3945

Linia 20 mówi nam wyraźnie od której komórki program się uruchamia; jego przeniesienie na taśmę powinno wyglądać następująco: 1. Wczytaj do pamięci program MASTER (lub podobny, używany przez Ciebie o którym wiesz, że działa jak w/w);

2. Uruchom go; następnie wykonaj NEW <RETURN>;

3. Wpisz do pamięci następującą linijkę: 10 SYS 3945 <RETURN>

4. Wczytaj główną część programu za pomocą LOAD „ABCD", 8, 1

5. Sprawdź wartości w komórkach 43-46; jeżeli otrzymałeś 1, 8, 3, 8 to musisz wrócić do punktu 2;

6. Jeżeli dwie ostatnie wartości są różne od podanych powyżej to możesz już zapisać taki program na taśmie.

Czasami zdarza się jednak, że adres początku programu wynosi np. 25000; wtedy masz do wyboru następujące możliwości:

— albo zapiszesz cały program w postaci jednego bloku, w skład którego wejdzie 23K zbędnej pamięci (co na pewno jest łatwiejsze, wydłuża jednak czas wczytywania programu);

— możesz także wczytać monitor języka maszynowego i próbować przenieść program bliżej początku pamięci co niestety nie zawsze będzie możliwe.

W praktyce zawsze jest bezpieczniej zapisać nawet bardzo dużą „dziurę" gdyż bez dokładnego zbadania programu nie można stwierdzić czy nie będzie on w trakcie działania z „dziury" tej korzystał.

W podobny sposób można „składać" programy, które mają zapisane na dysku oddzielnie sam program i oddzielnie sprite'y oraz dane np. muzyczne. Procedura przenoszenia ich na taśmę jest w zasadzie taka sama z tym, że najpierw należy stworzyć ze wszystkich części (oczywiście po sprawdzeniu ich adresów wczytywania) jeden blok, który następnie zapisujemy na wolnym dysku za pomocą SAVE „NAZWA" , 8, 1. Przy okazji dobrą metodą jest wgrywanie każdej części osobno i sprawdzanie wartości w komórkach 45 i 46 — w ten sposób otrzymamy informację, która z części określa koniec programu. Załóżmy, że nasze programy mają następujące adresy początkowe i końcowe:

część 1 — 3000 do 24000
część 2 — 32768 do 39000
część 3 — 26000 do 32767

W takiej sytuacji po zgraniu wszystkich części należy ustawić jako adres końca programu 39000; po ustawieniu adresu początku (na 2049) należy zapisać na dysku jeden blok od 2049 do 39000.

Odrębne zagadnienie stanowią programy, które mają krótki program maszynowy jako program wczytujący pozostałe części. Konieczne jest wtedy określenie adresu komórki pamięci, od której program się uruchamia; dla nieznających języka maszynowego jest to prawie niemożliwe tak więc pozwolimy sobie pominąć gdyż nie zaliczają się one do programów NIEKTÓRYCH. Z tego samego powodu pomijamy również programy samou-ruchamiające się. (Ted)

LISTER LISTER LISTER

Poniższy program, często wykorzystywany przez redakcję „Klanu Commodore", pozwala na uzyskiwanie wydruków o dowolnej szerokości.

Program może się przydać, gdy np. chcemy wydrukować listing programu na węższym papierze bądź tak, by zostawić z boku trochę miejsca.

Przygotowanie danych do drukowania odbywa się następująco: należy załadować program przeznaczony do listowania i napisać:

OPEN 1, 8, 5, „LISTING,S,W" : CMD 1 : LIST PRINT 1: CLOSE 1

Następnie należy wgrać i uruchomić LISTER. Program pyta o szerokość wydruku. Musi to być liczba w zakresie 1-80.

Oprócz listingów LISTER może posłużyć do wydrukowania zawartości dowolnego zbioru typu SEQ. Należy w tym celu zmienić nazwę „listing" w komendzie OPEN (linia 1010) na własną i uruchomić LISTER.

Uwaga: program przez cały czas korzysta ze stacji dysków. Używanie go z magnetofonem możliwe jest teoretycznie (należy zmienić parametr 8 na 1). ale bardzo niewygodne.

(M.S.)

```
1000 INPUT"▯SZEROKOSC▯▮,,Q
     WYDRUKU";S
1010 OPEN 1,8,5,
     "LISTING,S,R":OPEN2,4
1020 GET#1,A$:IFA$=CHR$(13)
     THENC=0
1030 C=C+1:IF ST<>0 GOTO
     1060
1040 PRINT#2,A$:
     IFC=STHENPRINT#2: C=0
1050 GOTO1020
1060 PRINT#2:CLOSE1:CLOSE2

READY.
```

COPY

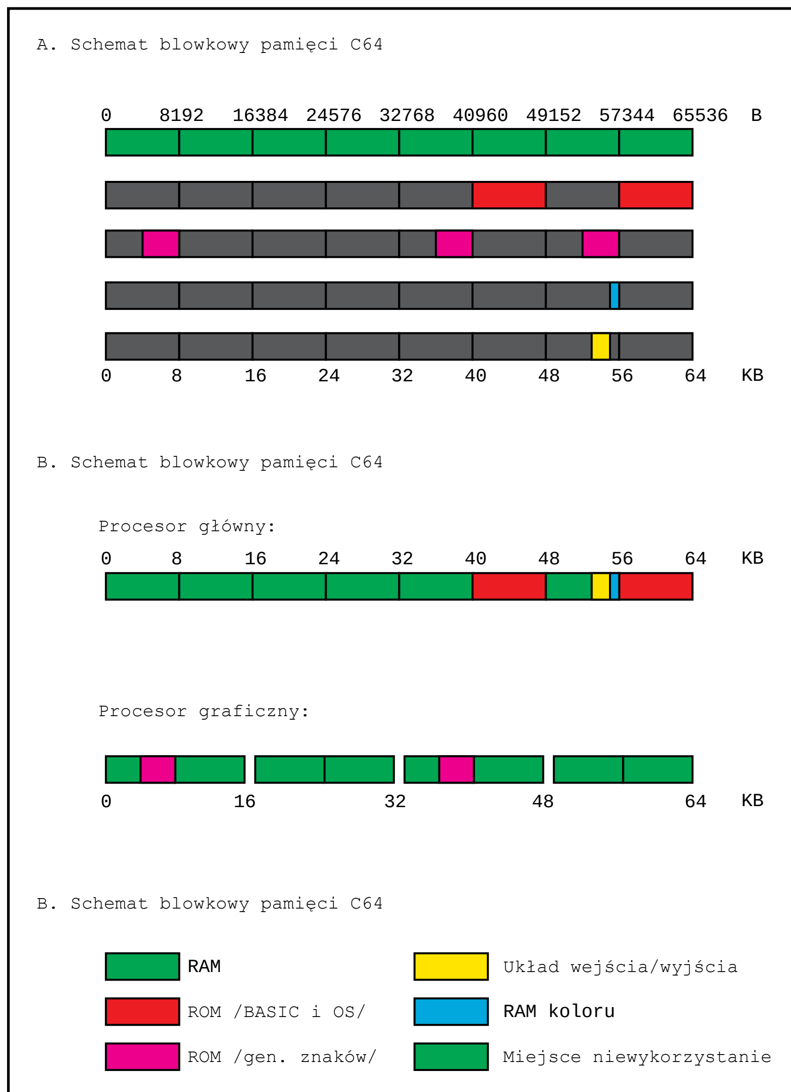
```
1000 DATA120,169,6,133,1,88,169,16,162,64,160,3,32,249,253,169
1001 DATA1,162,8,160,255,32,0,254,169,43,166,45,164,46,32,221
1002 DATA245,120,169,7,133,1,88,96
1003 FORJ=679 TO 718:READQ:POKEJ,Q:S=S+Q:NEXT
1004 IFS<>4411THENPRINT"POMY£KA W DANYCH !":STOP
1005 PRINT"PO WGRANIU PROGRAMU - SYS 679":NEW
READY.
```

TURBO. Nie ma jednak róży bez kolców, **TSL** daje możliwość kopiowania taśma-taśma, ale nic poza tym. Z kolei format zapisu używany przez **TSL** (nagłówek nagrany normalnie a reszta w TURBO) jest nieczytelny praktycznie dla wszystkich programów kopiujących. Tak więc przegranie programu z kasety na dysk jest — w przypadku programów dłuższych niz 39 K — niemożliwe. W tej sytuach nie pozostało mi nic innego, jak tylko napisać własny program kopiujący

Jego działanie jest proste, po uruchomieniu przez **SYS 679** na dysku zostaje zapisana zawartość pamięci RAM od adresu 2049 do (maksymalnie) 53247 Nazwa oraz adresy początku i końca programu są automatycznie wyczytywane z bufora magnetofonu. Przełączenie bloków pamięci powoduje, że zamiast interpretera języka BASIC na dysku zapisany zostanie zawarty „pod nim" w RAM program. Można kopiować programy o długości do 51 K.

Kopiowanie za pomocą **TSL COPY** odbywa się następująco: wczytujemy TSL COPY, potem program przeznaczony do skopiowania. Po włożeniu pustej dyskietki piszemy **SYS 679** i po chwili program jest skopiowany. Można teraz wczytać z taśmy następny program, napisać SYS 679 i tak w kółko. **TSL COPY** jest w pamięci do momentu wykonania procedury **RESET (SYS 64738)** lub, oczywiście, wyłączenia komputera.

TSL COPY można wykorzystać do innych celów. Wystarczy, że w adresy 832-848 wpiszemy nazwę, a w 43-46 adresy początku i końca, a po wykonaniu **SYS 679** dowolny obszar pamięci znajdzie się na dysku.



Rys. 1 Mapa pamięci C64

POLSKI ALFABET cz. II

Zgodnie z obietnica dana w poprzednim odcinku, dziś omówione będa sposoby przedefiniowania zestawu znaków. Na początek trochę teorii.

Za generację znaków i grafiki na ekranie C64 odpowiedzialny jest osobny, specjalizowany procesor wizyjny, zwany VIC (Video Interface Control-ler). Pracuje on z zegarem 8 MHz, może zaadresować 16 K pamięci, a sterowany jest za pośrednictwem 47 rejestrów w adresach 53248-53294. Jako przestrzeń pracy tego procesora może być wybrany dowolny 16 K bank w pamięci komputera. Normalnie procesor ustawiony jest na czytanie pierwszych 16 K, czyli banku O (adresy od 0 do 16383). Oprócz dostępu do wspomnianych 16 K, dla VIC-a zarezerwowany jest 1 K pamięci koloru w adresach 55296-56319. Pamięć ta wykorzystywana jest wyłącznie przy pracy w trybie tekstowym, a sposób jej wykorzystania można sprawdzić za pomocą prostego testu:

POKE 1024,1 — w lewym górnym rogu ekranu powinna pojawić się litera A

POKE 55296,C — litera zmienia kolor w zależności od wartości parametru C.

Ingerując w zawartość pamięci koloru można dowolnie zmieniać kolory tekstu, nie naruszając zawartości ekranu. W adresach 55296-56319 mieści się również ROM zawierający dane dla tworzenia kolorów, jednakże wykorzystanie go jest czysto sprzętowe i nie będziemy się nim zajmować.

W ramach swoich 16 K procesor graficzny przechowuje następujące dane:

— 1 K pamięci ekranu tekstowego
— 8 K pamięci ekranu graficznego
— od 0 do 16 K danych dla sprite'ow
— 4 K danych dla tworzenia znaków (tzw. generator znaków)

Rozmieszczenie tych obszarów w 16 K przestrzeni adresowej VIC-a jest dowolne. Ostatnie 8 bajtów pamięci ekranu tekstowego (przy normalnym ustawieniu procesora są to adresy 2040-2047) wykorzystane jest jako wskaźniki sprite'ow, czyli parametry informujące procesor, gdzie znajdują się w pamięci dane potrzebne do generacji sprite'ow.

Procesor graficzny w ramach „swoich" 16 K ma dostęp do innych bloków pamięci, niż procesor główny. Ilustruje to rys. 1

Skoro wiemy już, jak VIC zarządza pamięcią, warto teraz zająć się teorią powstawania znaków. Jak już wspomniałem, 4 K w przestrzeni adresowej procesora zarezerwowane są dla generatora znaków. W ich obrębie zawarte są dwa zestawy, jeden z nich zawiera duże litery i komplet znaków graficznych, drugi zaś litery duże, małe i tylko niektóre znaki graficzne. Zestaw przełączamy, naciskając jednocześnie klawisze SHIFT i COMMODORE LOGO. Pojedynczy zestaw znaków, zajmujący 2 K, zawiera dane dla grafiki typu mapy bitów. Na podstawie tych danych VIC generuje rysunki o kształcie zależnym od numeru znaku, zawartego w pamięci ekranu. Na dane dla jednego znaku składa się 8 bajtów. Dla tych, którzy nie zetknęli się jeszcze z grafiką „bitmap", zamieszczam krótki opis postępowania przy projektowaniu własnych znaków.

Jeden bajt składa się z ośmiu bitów. Każdy bit może przyjmować wartość 0 lub 1. Graficzne przedstawienie bajtu o wartości 44 wygląda następująco:

0 0 1 0 1 1 0 0

Ponieważ poszczególne bity, licząc od prawej do lewej strony, zgodnie z prawami zapisu pozycyjnego mają wartości odpowiadające kolejnym potęgom dwójki, czyli odpowiednio 1, 2, 4, 8, 16, 32, 64 i 128 nasz przykładowy bajt można przedstawić również w taki sposób:

00 + 00 + 32 + 00 + 08 + 04 + 00 + 00

Po zsumowaniu wartości bitów ustawionych na 1 otrzymujemy wartość 44 = (4 + 8 + 32) Na rysunku 2 po lewej stronie, widzimy literę A ro-zrysowaną w siatce 8 x 8 Po prawej stronie ta sama litera przedstawiona jest w postaci bajtów i przeliczona na wartości dziesiętne:

Rysując w siatce literę A, a następnie sumując wartości ustawionych bitów, uzyskaliśmy szereg liczb: 8, 28, 54, 99, 127, 99, 99, 0. Analogicznie projektujemy pozostałe znaki. Uzyskany dwu kilobajtowy zestaw danych umieszczamy w pamięci. Procesor wizyjny, wyświetlając jakiś znak, odwoła się do umieszczonych przez nas w pamięci wzorów, pobierze osiem kolejnych bajtów i wyświetli na ekranie te punkty, które odpowiadają bitom ustawionym na 1. W rezultacie na ekranie pojawi się żądany znak.

Dzięki opisanej technice możemy już projektować własne znaki. Został jednak do rozwiązania problem znalezienia adresu w obrębie matrycy znaków, pod którym będziemy umieszczać dane. Chodzi o to, aby naciśnięcie klawisza z wizerunkiem danej litery powodowało wyświetlenie tego właśnie, a nie innego znaku. Rozwiązanie polega na użyciu formuły:

ADR = MAT + NR x 8

gdzie ADR to szukany adres, MAT — adres początku matrycy znaków, zaś NR to numer znaku wzięty z tabeli kodów ekranowych (patrz rys. 3). Tabela ta zawiera numery znaków w kolejności ich występowania w generatorze (matrycy). Przykładowo, jeśli projektujemy literę A, po narysowaniu jej w siatce 8 x 8 i wyliczeniu ośmiu bajtów, znajdujemy w tabeli numer litery A i podstawiamy do wzoru. Zakładam, że generator znaków umieszczony jest w pamięci od adresu 12288:

ADR = 12288 + 1 x 8 = 12296

Widzimy, że dane dla litery A należy wpisać w komórki pamięci o adresach 12296-12303. Litera B będzie wpisana odpowiednio w adresy 12304-12311 itp.

Opisany sposób projektowania znaków daje duże możliwości nie tylko w przypadku projektowania liter. Program zamieszczony na końcu tego artykułu pokazuje próbkę możliwości tej techniki.

Wracając jednak do sprawy polskich liter. Jeśli chcemy zmienić tylko kilka znaków, czy musimy w tym celu projektować od nowa wszystkie? Na szczęście nie, istnieje sposób na przepisanie zawartości matrycy znaków z ROM-u do RAM-u. Potem wystarczy tylko wprowadzić drobne zmiany i nowy zestaw znaków mamy gotowy. Program przepisujący znaki wygląda następująco:

```
10 MAT = 12288
20 POKE 56333,127 : POKE 1,PEEK (1) AND 251
30 FOR X = 0 TO 2048
40 POKE MAT + X, PEEK (53248 + X)
50 NEXT X
60 POKE 1,PEEK (1) OR 4 : POKE 56333,129
```

Linia 10 ustala adres generatora znaków w RAM. Linia 20 wyłącza przerwania NMI i przełącza bloki pamięci. Linie 30, 40 i 50 tworzą pętlę przepisującą pierwszy zestaw znaków (2 K), który po przełączeniu pamięci pojawia się w adresach 53248-55295. Można za jednym zamachem przepisać oba zestawy znaków, należy wtedy zmienić linię 30 następująco:

```
....X...  bajt 0, 00001000:  8
...XXX..  bajt 1, 00011100:  4+8+16=28
..XX.XX.  bajt 2, 00110110:  2+4+16+32=54
.XX...XX  bajt 3, 011000110: 1+2+32+04=99
.XXXXXXX  bajt 4, 011111111: 1+2+4+8+16+32+64=127
.XX...XX  bajt 5, 01100011;  1+2+32+64+99
.XX...XX  bajt 6, 01100011;  1+2+32+64=99
........  bajt 7, 00000000;  0
```

Rys. 1 Mapa pamięci C64

```
DEFINIOWANIE ZNAKOW

1000 PRINTCHR$(147);CHR$(142);CHR$(8):POKE56333,127:POKE1,PEEK(1)AND251
1010 FORX=0TO2048:POKE12288+X,PEEK(53248+X):PRINTX,CHR$(19);:NEXT
1020 POKE1,PEEK(1)OR4:POKE56333,129:POKE53272,(PEEK(53272)AND240)+12
1030 PRINTCHR$(147):FORA=0TO5:FORB=0TO39:PRINTCHR$(64+A);:NEXT:PRINT:NEXT
1040 READN:FORX=0TON*8-1:READQ:POKE12288+X,Q:NEXT:PRINT
1050 FORA=1TO80:PRINT"F";:FORB=0TO7:POKE12288+6*8+B,RND(1)*256:NEXT:NEXT
1200 DATA 6
1300 DATA 255,129,129,129,129,129,129,255:      REM PROGRAM TEN POZWALA NA
1310 DATA  28, 54, 99, 99,127,99,99,6:          REM PRZEDENIOWANIE ZESTAWU
1320 DATA 126,129,165,165,129,126,36,102:       REM ZNAKOW.ZMIENIONO ZNAKI
1330 DATA  24, 36, 36, 40, 36, 36,24,192:       REM O KODACH ASCII OD 64 DO
1340 DATA   3,  5,  9, 17, 31, 33,65,129:       REM 70, CZYLI KODACH EKRANO-
1350 DATA 255,129,189,165,165,189,129,255:      REM WYCH OD 0 DO 6.
READY.
```

```
ZMIENIONY KURSOR
1020 POKE1,PEEK(1)OR4:POKE56333,129:POKE53272,(PEEK(53272)AND240)+12
1030 PRINTCHR$(147):FORA=0TO5:FORB=0TO39:PRINTCHR$(64+A);:NEXT:PRINT:NEXT
1040 READN:FORX=0TON*8-1:READQ:POKE12288+X,Q:NEXT:PRINT
1050 FORA=1TO80:PRINT"F";:FORB=0TO7:POKE12288+6*8+B,RND(1)*256:NEXT:NEXT
1200 DATA 6
1300 DATA 255,129,129,129,129,129,129,255:      REM PROGRAM TEN POZWALA NA
1310 DATA  28, 54, 99, 99,127,99,99,6:          REM PRZEDENIOWANIE ZESTAWU
1320 DATA 126,129,165,165,129,126,36,102:       REM ZNAKOW.ZMIENIONO ZNAKI
1330 DATA  24, 36, 36, 40, 36, 36,24,192:       REM O KODACH ASCII OD 64 DO
1340 DATA   3,  5,  9, 17, 31, 33,65,129:       REM 70, CZYLI KODACH EKRANO-
1350 DATA 255,129,189,165,165,189,129,255:      REM WYCH OD 0 DO 6.
READY.
```

30 FOR X = 0 TO 4096

linia 60 przywraca początkowy układ bloków pamięci i na powrót włącza przerwania. Dla wyjaśnienia: NMI, czyli przerwania niemaskowalne, pełnią w komputerze dosyć ważną rolę. Przełączenie bloków pamięci, w których przerwania te działają, byłoby jak wyjęcie autostrady spod pędzących samochodów. Skutek łatwo sobie wyobrazić. Dlatego też przed wszelkimi operacjami naruszającymi podstawowe funkcje komputera przerwania należy wyłączyć.

Po wykonaniu powyższego programu kopia generatora znaków znajduje się w pamięci RAM w adresach 12288-14335 lub 12288-16383 (zależnie od tego, którą wersję linii 30 komputer wykonał). Niestety, procesor graficzny o tym nie wie i dalej korzysta ze swojego ROM-u. Sposób na poinformowanie VIC-a o lokalizacji nowego zestawu znaków w pamięci jest następujący:

POKE 53272, (PEEK (53272) AND 240) + ADR/1024

Pod adresem 53272 mieści się jeden z rejestrów sterujących procesora wizji, zwany MCR (Memory Control Register, Rejestr Sterowania Pamięcią). Bity 0-3 w tym rejestrze przechowują wskaźnik lokalizacji matrycy znaków. Jest to liczba parzysta w zakresie 0-14, która pomnożona przez 1024 daje adres matrycy znaków, czyli wartość naszej zmiennej ADR. Z takiego sposobu przechowywania tej liczby wynika powyższy wzór, jak również fakt, iż zestaw danych dla znaków może zaczynać się tylko od adresu podzielnego przez 2048.

Po ustawieniu procesora graficznego na czytanie matrycy znaków z RAM-u możemy zacząć eksperymenty. Należy znaleźć takie klawisze, którym można przypisać polskie litery bez utraty znaków ważnych przy używaniu komputera, jak dwukropek czy średnik. Następny artykuł zawierać będzie propozycję optymalnego rozmieszczenia znaków na klawiaturze, jednakże mimo to zachęcam wszystkich do prób.

Na zakończenie program demonstrujący technikę wprowadzania własnych znaków. Nie będę opisywał działania każdej linii, ponieważ te same linie zostały już wcześniej omówione w tekście. Na uwagę zasługuje tylko część demonstracyjna. Zmienione jest pierwsze sześć znaków, o kodach ekranowych 0-5, kodach ASCII 64-69. Siódmy z kolei znak (litera F) został wykorzystany do pokazania możliwości zmiany kształtu znaku w czasie. Powrót do starego zestawu znaków odbywa się przez naciśnięcie klawiszy RUN/STOP i RESTORE. Polecam eksperymenty, polegające na „ręcznym" — czyli w bezpośrednim trybie pracy — wpisywaniu różnych wartości w obręb matrycy znaków i obserwowanie efektów. Proponuję taki eksperyment:

FOR X = 12288 TO X + 2048 : POKE X, RND (1) - 256 : NEXT

Zaręczam, że efekt będzie interesujący, oczywiście w sensie poznawczym, bo pracować z tak uzyskanym zestawem znaków byłoby raczej trudno. Następny odcinek zawierać będzie przykłady konkretnej realizacji polskich znaków za pomocą zmiany matrycy.

M.S.

Nie podajemy całej tabeli, ponieważ:

- **a. znaki o kodach od 128 do 255 to po prostu „negatywy" pierwszych 128 znaków**
- **b. drugi zestaw znaków, jako zawierający zarówno duże, jak i małe litery lepiej nadaje się do naszych celów**
- **c. pełną tabelę można znaleźć w prawie każdej instrukcji obsługi**

TABELA KODÓW EKRANOWYCH DRUGI ZESTAW ZNAKÓW

0	@	1	a	2	b	3	c	4	d	5	e
6	f	7	g	8	h	9	i	10	j	11	k
12	l	13	m	14	n	15	o	16	p	17	q
18	r	19	s	20	t	21	u	22	v	23	w
24	x	25	y	26	z	27	[	28	‾	29	]
30	↑	31	←	32		33	!	34	"	35	#
36	$	37	%	38	&	39	'	40	(	41	)
42	*	43	+	44	,	45	–	46	.	47	/
49	0	49	1	50	2	51	3	52	4	53	5
54	6	55	7	56	8	57	9	58	:	59	;
60	<	61	=	62	>	63	?	64	–	65	A
66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M
78	N	79	O	80	P	81	Q	82	R	83	S
84	T	85	U	86	V	87	W	88	X	89	Y
90	Z	91	┼	92	▌	93	│	94	╳	95	▒
96		97	▌	98	▄	99	▔	100	▁	101	▏
102	▒	103	▕	104	▚	105	▧	106	▕	107	├
108	▗	109	└	110	┐	111	▁	112	┌	113	┴
114	┬	115	┤	116	▎	117	▍	118	▐	119	▔
120	▀	121	▄	122	┘	123	▖	124	▝	125	┘
126	▘	127	▚								



YOUR COMMODORE

Wydawane od połowy 1984 roku roku, o przeciętnej objętości 80 stron czasopismo, nie przynosi na swoich łamach nic nowego, czego by nie można było znaleźć w innego tego typu periodykach. Obok opisu sprzętu i urządzeń peryferyjnych, znajdujemy recenzje gier i oprogramowania użytkowego. Od czasu do czasu redakcja przeprowadza zbiorowe testy porównawcze ostatnich modeli drukarek, modemów, piór świetlnych, tabliczek cyfrowych i innego rodzaju sprzętu, stworzonego oczywiście z myślą o C 64/128. A jest tego całe mnóstwo: przeglądając, nawet pobieżnie, jakikolwiek numer YC, jeszcze raz można się przekonać, iż C 64 jest chyba najlepiej oprzyrządowanym komputerem domowym świata!

Ciekawe są też artykuły poświęcone zastosowaniem praktycznym Commodore 64, np. dostosowaniem go do współpracy z różnymi instrumentami elektronicznymi.

Innym, interesującym działem, jest cykl artykułów poświęconych językom programowania, np. „Jak opanować kod maszynowy" czy „Rozszerz swój BASIC".

Wszystkie opisywane urządzenia oraz zamieszczone wydruki programów są publikowane głównie z myślą o modelach C 64 i VIC 20. Na kilka przejrzanych numerów, tylko w jednym z nich znaleźliśmy program dla C 116. O modelach +4 i 16 nawet nie wspomniano. Od pewnego natomiast czasu YOUR COMPUTER drukuje programy napisane dla modelu C 128, przy czym — najczęściej — wykorzystywana jest tu wysokorozdzielcza grafika trybu 128.

Pismo nie jest tanie, 90 szylingów. Należy więc żałować, iż za tę cenę i przy tej jakości papieru, kolorowe strony można policzyć na palcach jednej ręki, a i te reklamowały raczej mało ciekawe artykuły. Natomiast żadna z 17 recenzowanych gier nie miała kolorowego zdjęcia!

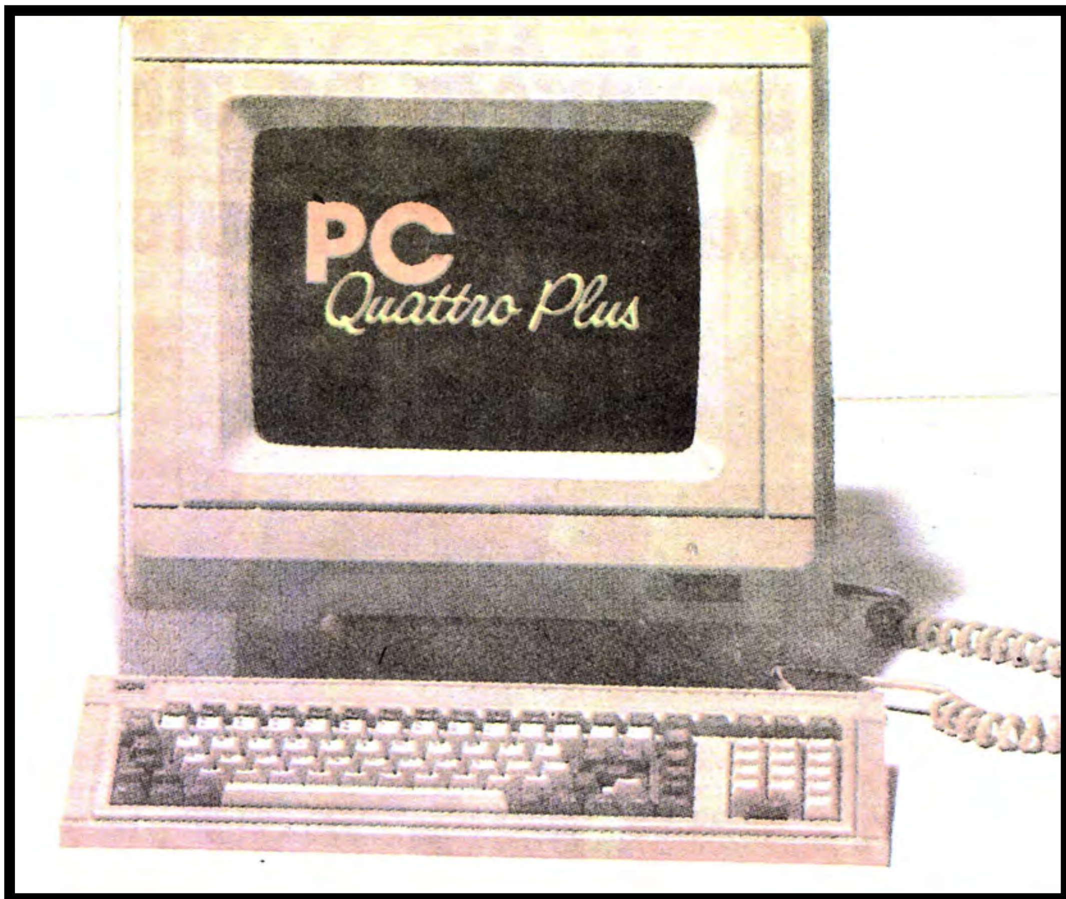
Tak więc uwaga zwolennicy Commodore. Zastanówcie się zanim wydacie na prenumeratę blisko 17 funtów. Czy macie jakiś wybór? Owszem, w Wielkiej Brytanii jest jeszcze siedem innych tytułów czasopism wydawanych z myślą o posiadaczach różnych modeli Commodore, jedne bardziej kolorowe od drugich. Ale zawsze pozostaje Wam ... Bajtek. I to za jedyne 100 złotych!

YOUR COMMODORE
No 1 Golden Square
LONDON WIR 3AB, ENGLAND

Jerzy Zawadzki



ATARI 130 XE
Q W E R T Y
Pewex
ATARI®
KOMPUTER POD CHOINKĘ



BALTEXPO ’86
KOMPUTERY ICL

Już po raz trzeci producenci sprzętu morskiego spotkali się we wrześniu tego roku w gdańskiej Hali „Olivia”. Wystawa Baltexpo'86, z roku na rok gromadzi coraz więcej wystawców.

Nie ma ona charakteru handlowego, co jednak nie wyklucza podpisywania kontraktów czy też prowadzenia rozmów wstępnych.

Wśród licznych wystawców nasze zainteresowanie wzbudziła największa brytyjska firma komputerowa International Computers Limited. Była to jedyna firma prezentująca systemy komputerowe na poziomie światowym.

Dyr. JAN KLUK mówiąc o owocach współpracy International Computers Limited z Polską powiedział: — **Na rynku polskim istnieje ponad 400 instalacji komputerowych ICL i pochodnych. Podpisaliśmy kontrakty m.in. z przemysłem stoczniowym w Gdańsku, Szczecinie i Ustce; z przemysłem ciężkim (Huta im. Lenina, Zakłady Metalowe Cegielskiego); centralami handlu zagranicznego: Polservice, Ciech, Skórimpex, Textilimpex, Cenfexim, Polmot; przedsiębiorstwami transportowymi: PKP, PKS oraz wieloma innymi.**

Filozofię ICL dyr. KLUK przedstawił w następujących słowach: — **My chcemy współpracować z rynkiem a nie sprzedawać i dlatego interesuje nas sprzedaż nie skrzynek** (tak wyglądały te najnowocześniejsze elementy) **a całych systemów w pełni odpowiadających konkretnym wymaganiom**.

Nie ograniczamy się do sprzedaży. Komputer jest pojazdem — ważne jest również dokąd jedziemy. Dlatego przedstawiamy koncepcję na następne 10 lat.

Potwierdzeniem słów Dyrektora Oddziału w Polsce J. Kluka był przedstawiony przez ICL sprzęt komputerowy. Szczególne zainteresowanie wzbudzał system mikrokomputerowy **DRS 300** będący wg zapewnień pracowników ICL szczytowym osiągnięciem technologii. Daje możliwość budowania systemu lub sieci komputerowej w bardzo krótkim czasie z modułów nie większych niż biurowe segregatory.

Na wystawie został pokazany również reprezentant rodziny mikrokomputerów **O’** serii **QUATTRO.** Dowiedzieliśmy się, że pojawił się już **PC QUATTRO PLUS**. Niestety, do obejrzenia jedynie nawet jeżeli nie oznacza to, że na **BALTEXPO'86** firma ICL przygotowała zestaw anachroniczny. Jest jedynie oznaką tempa rozwoju i kolosalnych przyspieszeń w technice mikrokomputerowej, **QUATTRO PLUS** po prostu spóźnił się na pociąg do Gdańska.

O znaczeniu jakie przywiązuje firma ICL do wystawy **BALTEXPO’86** najlepiej świadczy zaprezentowany, wg jej przedstawicieli najnowocześniejszy na świecie pokładowy system komputerowy kontroli załadowania dla morskich statków towarowych. Ma ona zapewnić optymalną eksploatację statku i, co najważniejsze, bezpieczeństwo. Przy jego pomocy możliwa jest stała kontrola stateczności statku i towaru. System **LOCON** wydaje dyspozycje załadunku i rozładunku statku, pozwala na natychmiastową ich modyfikację oraz wariantową analizę stanu załadowania. Może się również zajmować kontrolą: stateczności awaryjnej, obciążeń pokładów, obciążeń skręcających zbiornikowców. System ten może też mieć połączenie teletransmisyjne z dyspozytorami portowymi i armatora.

System **LOCON** dostarcza informacji dotyczących ewentualnego przeładowania towaru dla zachowania bezpieczeństwa. Może również prowadzić analizy ekonomiczne tzn. jaki towar skąd i dokąd ma być przewieziony, a także jaka jest cena usługi frachtowej.

Podobno ta absolutna pewność ma być zainstalowana na polskich statkach. Miejmy nadzieję, że do tego dojdzie. Wprowadzenie nowoczesnych systemów mikrokomputerowych na statki **PLO** i **PŻM** jest koniecznością nie tylko chwili ale również i sprostania wymaganiom stawianym przez towarzystwa ubezpieczeniowe. Nie można więc długo zwlekać z zakupem, pamiętając jednak, że kupujący ma prawo wyboru najlepszego towaru.

Dariusz Dewille

MIKRODATOR CZYLI KOMPUTER PO SZWEDZKU

KORESPONDENCJA ZE SZWECJI

Przyjeżdżającemu do Malmo natychmiast rzuci się w oczy niezwykła czystość panująca w tym mieście i olbrzymia ilość sklepów od zawartości których można dostać zawrotu głowy. Niestety znacznie częściej można też dostać zawrotu od cen tu panujących — Szwecja jest raczej drogim krajem.

Na pierwszy rzut oka komputerów specjalnie nie widać, jeśli nie liczyć automatów bankowych czy reklam na wystawach sklepów. Dopiero po pewnym czasie zaczyna się dostrzegać małe sklepiki i luksusowe supermarkety, na wystawach których niepodzielnie króluje małe plastikowe pudełko z klawiaturą. Jak to zwykle bywa firma sprzedająca czy to komputery czy oprogramowanie oferuje inne udogodnienia — nie ma co ukrywać, w porównaniu z cenami w RFN panuje tu drożyzna.

Sprzętu jest dosłownie masa. Najbardziej popularnymi komputerami są tu modele firmy COMMODORE; popularnymi do tego stopnia, że trzeba czasami tydzień czekać zanim sprzęt nie zostanie ściągnięty z hurtowni. Dotyczy to zwłaszcza modelu nr 1 na szwedzkiej liście przebojów — COMMODORE 128D. Z siedmiu sklepów, w których pytałem o ten model, był on tylko w jednym i to też dwie ostatnie sztuki.

W zasadzie brak tu takich komputerów jak COMMODORE 16, COMMODORE 116, czy VIC 20. Bardzo tanie sprzedaje się COMMODORE +4, który jest tu w cenie od 36 (!) do 100 dolarów. Wszelkich innych modeli od nowej wersji COMMODORE 64/64 C — wyposażona w nową klawiaturę, nową obudowę od C 128 i dodatkowy ROM z programem GEOS/ do AMIGI włącznie jest w sklepach sporo.

Oprócz Commodore najbardziej eksponowanym w sklepach jest komputer VICTOR — kompatybilny z IBM PC/AT i oferujący podobne możliwości i parametry techniczne. Z „drobnicy” można spotkać też ZX SPECTRUM (na wymarciu definitywnym), kilka komputerów MSX i nieliczne, pojedyncze egzemplarze Amstrada CPC 464. W dużych profesjonalnych sklepach można kupić lub wynająć taki sprzęt jak Hewlett PACKARD, ERIKSSON, IBM, ICL, PDP, itp. Są to już komputery profesjonalne, tak więc i młodych entuzjastów w sklepach tych nie widać — ceny robią swoje.

Nigdzie nie mogłem trafić na najmniejszy choćby ślad firmy ATARI — sprzedawcy twierdzili, że jest to sprzęt mało popularny i nie opłaca sie go sprzedawać.

Na zakończenie chciałbym wspomnieć o czasopismach, gdyż dla znających szwedzki są one dostępne w każdym większym supermarkecie. Anglojęzyczne wersje COMPUTE! RUN COMMODORE PO-WER/PLAY/ COMPUTE! SINCLAIR USER, MSX COMPUTERS, itp. można kupić za kilka dolarów (średnio ok. 4-5 za czasopismo) w „International Press Center” przy placu Gustawa Adolfa. Są tam także niemieckie magazyny jak 64'er, MC itp.

Najsmutniejszym widokiem są polskie napisy w sklepach, dość wymownie świadczące o „chlubnej” opinii jaką pozostawili po sobie niektórzy rodacy „turyści”. Dotyczy to także sklepów z komputerami, gdzie na dźwięk polskiej mowy sprzedawcy niechętnie wypożyczają dyskietkę czy kasetę do obejrzenia programu — zdarzały się bowiem wypadku, że ulatniały się one wraz z „klientem”.

Klaudiusz Dybowski

Uwaga ... Start!!!



RADIOKOMPUTER

Te zapowiedź zna już część słuchaczy programu IV Polskiego Radia. Poprzedza ona emisję programów komputerowych w nadawanej od 1 lipca br. audycji pt. Rozgłośni Harcerskiej pt. „RADIOKOMPUTER".

Już dwa lata temu, w wyniku rosnącego zainteresowania słuchaczy mikrokomputerami i problemami związanymi z informatyką, zacząłem emitować programy komputerowe w czwartkowym magazynie naukowo-technicznym Rozgłośni Harcerskiej.

Po pierwszych emisjach okazało się, że programy odbierane na zwykłych odbiornikach radiowych z zakresem fal UKF, nagrywane na magnetofon, wczytują się poprawnie do komputerów (dokładnie ok. 70 proc). Pojawiło się więc nowe źródło oprogramowania dostępnego dla wszystkich chętnych w całej Polsce. Dodajmy — dostępnego ZA DARMO! Tego typu emisje, prowadzone przez Rozgłośnię Harcerską, były pierwszymi w bloku państw socjalistycznych.

W odpowiedzi na prośby słuchaczy zainteresowanych tą formą wymiany doświadczeń i oprogramowania, pod patronatem Rozgłośni Harcerskiej powstał „RADIO-KOMPUTER". Do współpracy zaprosiłem Krzysztofa Surgowta i Piotra Tymochowi-cza, autorów telewizyjnej audycji „Hallo — Komputer". Chcemy pokazać komputer, jako narzędzie pomocne w pracy, nauce i domu. Dlatego w RADIOKOMPUTERZE emitowane są programy edukacyjne i użytkowe (IQ TEST, Beta Basic 3.0, The Artist, Poltasword, Matematyka, mikroProlog i in.), będące ripostą na falę gier telewizyjno-komputerowych.

Naszym zamierzeniem jest również stymulowanie aktywności młodych ludzi do tworzenia nowych wartościowych programów. Temu zadaniu podporządkowane są ogłaszane w RADIOKOMPUTERZE konkursy. Przykładem jest wrześniowy konkurs na najciekawszy, użytkowy program autorski, w którym nagrodą był profesjonalny komputer LASER 700.

Na tym jeszcze nie koniec! RADIOKOMPUTER to także porady, oryginalne pomysły i najnowsze informacje ze świata informatyki. RADIOKOMPUTER czeka na dalsze sugestie słuchaczy co do formy I treści audycji.

W wielu listach pojawiają się problemy związane z nagrywaniem emitowanych przez radio programów i ich wczytywaniem do komputerów. Korzystając więc z łamów „BAJTKA" chciałbym podać naszym słuchaczom kilka rad:

• Jako nośniki pamięci zewnętrznej taśmowej należy stosować kasety typu Compact z taśmą żelazową (Fr.). Taśma nie może być pogięta i starta. Nie zalecam stosowania taśm produkcji polskiej (Stilon).

• Programy powinny być zapisywane na magnetofonie monofonicznym. Jeżeli dysponujecie magnetofonem stereofonicznym nagrywać należy na oba kanały, ale odtwarzać już tylko z jednego. Nagrywany sygnał można przesterować.

• Nagrywamy i odtwarzamy na tym samym magnetofonie.

• Szczelina głowicy w magnetofonie musi być prostopadła do kierunku przesuwu taśmy magnetycznej w kasecie. Skos głowicy można ustawić wykorzystując takie programy jak „Tape Head", „Tapper".

• Głowica nagrywająco-odtwarzająca powinna być czysta. Głowicę zabrudzoną drobinami warstwy czynnej taśmy magnetycznej przemywamy bawełnianą szmatką lub watą nasączoną spirytusem (eterem).

• Wszelkie układy redukcji szumów (Dolby Nr, CNRS-2, DBX) muszą być wyłączone.

• Sygnał radiowy powinien być czysty, pozbawiony szumów i przydźwięku (wskaźnik dostrojenia max). Konieczne jest zainstalowanie odbiorczej anteny zewnętrznej, najlepiej dipolowej. Wskazane jest uziemienie odbiornika radiowego.

• Wszystkie kable łączące radio-magnetofon-komputer muszą być ekranowane.

• Barwa głosu wpływa na poprawność wczytywania programów do komputera. W przypadku wystąpienia kłopotów należy magnetofon podłączyć do wzmacniacza mocy. Wykorzystując wyjście głośnikowe oraz regulując barwę głosu (podbicie, obcięcie wysokich lub niskich tonów) można zlikwidować kłopoty występujące przy wczytywaniu programów (oczywiście przy spełnieniu innych wymaganych warunków).

• Na czas trwania emisji należy wyłączyć w domu niepotrzebne urządzenia elektryczne (odkurzacz, pralka, zapalniczka do gazu).

Przypominamy: RADIOKOMPUTER to audycja nadawana w każdy piątek, o godzinie 16.00, program IV, Polskie Radio.

Nasz adres: **Rozgłośnia Harcerska 00-950 Warszawa, ul. M. Konopnickiej 6, dopisek „RADIOKOMPUTER".**

Tomasz Jordan, lat 28
absolwent PW



Naszym hobby od kilku lat jest wszystko co wiąże się ze słowem „byte". Okazało się, że nie byliśmy na osiedlu wyjątkami, a dzieciaki na podwórku coraz częściej obradowały, jak rozbroić „Night Gunnera", a jak być nieśmiertelnym w „Knight Lore". W konsekwencji we wrześniu ub. roku jako placówka Działu Społeczno-Wychowaw-czego SBM „Politechnika", powstał Klub Mikrokomputerowy HOBBYTE.

Na początek dostaliśmy zestaw ze Spectrum Plus i drukarką GP-50S. Zaadaptowano też stosowne pomieszczenie ze starej suszarni. Dziś mamy 3 pełne zestawy Spectrum, prowadzimy zajęcia 3 razy w tygodniu w 6-ciu kilkuosobowych grupach. Naszym celem obok popularyzacji jest nauczanie młodych pasjonatów nie tylko jak uniemożliwić postronnym wylistowanie swojego programu, ale też jak np. rozwiązać układ n równań liniowych. Program na ten temat, publikowany w tym numerze powstał na klubowych zajęciach.

Staramy się, by zaawansowanie uczestnika nie szło w kierunku programowania w asemblerze czy w kodzie maszynowym lecz raczej, jak to się mądrze nazywa, w kierunku metod numerycznych.

Uczestnictwo w zajęciach jest odpłatne, ale po każdych następują 2 godz. tzw. otwarte — wszyscy zainteresowani mogą przyjść, poradzić się, czy też w coś zagrać. Jesteśmy zainteresowani wszelką wymianą doświadczeń, pomysłów, informacji ze wskazaniem na oprogramowanie i praktycz ne wykorzystanie mikrokomputerów.

Siedziba Klubu: Warszawa, ul. Malawskiego 7, XI p. (piątek, sobota).

Filia: Warszawa, ul. Chodkiewicza 4, przyziemie (piątek).

Adres do korespondencji:
SBM „Politechnika"
ul. Etiudy Rewolucyjnej 48
02-643 Warszawa,
Dział Społeczno-Wychowawczy
„HOBBYTE"
tel.48-54-11 wewn. 61.

Krzysztof Herman
(instruktor)

FORUM ATARI

Z inicjatywy Przedsiębiorstwa Zagranicznego Karen i Ośrodka Doskonalenia Kadr Technicznych RS NOT zostało utworzone Studium-Forum Mikrokomputerowe Atari.

Podstawowym celem działania Forum jest organizowanie i udzielanie pomocy szkoleniowej i informacyjnej użytkownikom mikrokomputerów Atari w zakresie sprzętu, oprogramowania i zastosowań.

Dla osiągnięcia tych celów Forum organizuje pokazy, seminaria tematyczne, kursy oraz inne formy szkoleniowe, prowadzi bibliotekę poradników, dokumentacji i publikacji dotyczących mikrokomputerów Atari, prowadzi i udostępnia uczestnikom bibliotekę niezastrzeżo-nych programów i pakietów standardowych.

Jako pierwszy uruchomiony został kurs „Podstaw programowania w języku BASIC".

Przed oficjalnym rozpoczęciem działalności Forum zostały zorganizowane w miesiącach czerwiec — lipiec seminaria tematyczne poświęcone zapoznaniu ze sprzętem i oprogramowaniem.

Wydaliśmy również dla potrzeb szkoleniowych poradnik pod redakcją Wiesława Miguta pt. „ATARI — BASIC. Język programowania i obsługa mikrokomputerów Atari".

Efektem naszych dotychczasowych działań są również dwa programy: „Turbo - Copier 3/4" oraz CCD (program kopiujący w relacji kaseta — dysk).

Uczestnictwo w działalności Forum może mieć charakter stały lub tylko związany z konkretnym przedsięwzięciem (tzn. udział w seminarium, kursie itp.).

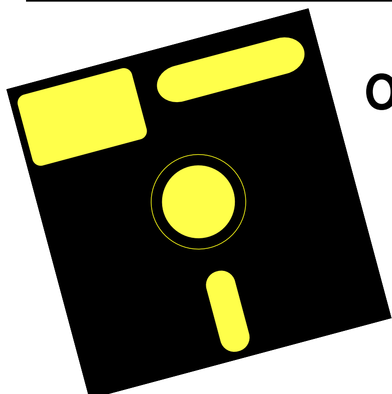
Uczestnictwo o stałym charakterze powstaje poprzez złożenie deklaracji, oraz opłacenie składki członkowskiej (za okres półroczny). Uczestnicy mogą korzystać ze stałego punktu konsultacyjnego, brać udział w organizowanych formach wymiany oprogramowania i literatury.

Deklarując przystąpienie do Forum każdy uczestnik przyjmuje na siebie zobowiązanie, że w ramach Forum nie będzie prowadził jakiejkolwiek działalności handlowej.

Inną formą działalności Ośrodka Doskonalenia Kadr Technicznych RS NOT w obszarze zastosowań mikrokomputerów jest Koło Użytkowników Mikrokomputerów Profesjonalnych. Koło to zrzesza jednostki gospodarcze będące użytkownikami mikrokomputerów typu profesjonalnego.

Siedzibą Forum Mikrokomputerowego Atari oraz Koła Użytkowników Mikrokomputerów Profesjonalnych jest Ośrodek Doskonalenia Kadr Technicznych RS NOT w Warszawie, ul. Przemyska 11a, telefon: 22-23-31 lub 22-77-23.

Robert Kamiński



KONKURS O ZŁOTĄ DYSKIETKĘ „BAJTKA"

W poprzednim numerze nasza redakcja ogłosiła współzawodnictwo klubów mikrokomputerowych O ZŁOTĄ DYSKIETKĘ „BAJTKA". Konkurs będzie okazją do wymiany doświadczeń w organizowaniu ruchu popularyzującego informatykę. Nie stawiamy żadnych ograniczeń, zapraszamy wszystkie kluby, także I te nieformalne. Za miesiąc opublikujemy kupon zgłoszenia do konkursu.

Bajtek

INDYWIDUALNY BANK DANYCH

Mateusz Bartosik lat 14, adres: ul. Koście lna 10/2, 46-081 Dobrzeń Wielki (woj. opolskie). Mam: komputer VG Philips (MSX) Moje zainteresowania to: informatyka, matematyka, numizmatyka.
Dariusz Filonowicz lat 17, ul. Monopolowa 8/53, 15-202 Białystok. Amstrad 6128. Informatyka, elektronika, motoryzacja, muzyka, sport, biotechnologie, technologie lotnicze i kosmiczne.
Tomasz Stańczyk lat 23, ul. Kościuszki 79/7, 50-442 Wrocław. Commodore C-128 D. Informatyka, sport, teatr, muzyka, język angielski, i jeszcze trochę.
Jacek Matuszczyk lat 23. ul. Darłowska 5A/15, 04-091 Warszawa, ZX Spectrum 48K. Krótkofalarstwo, fotografika, elektronika, informatyka, chemia.
Michał Jóźwiak lat 23, ul. Bandurskiego 216, 14-200 Iława. Sharp MZ-721. Informatyka (grać też lubię!).
Waldemar Strążyński lat 15, ul. Powstań ców Wlkp. 3/2, 64-360 Zbąszki (woj. zielo nogórskie). Sinclaire ZX 81 64K. Informatyka, systemy komputerowe.
Marek Horski lat 25, ul. Chorzowska 17/ 19, 44-100 Gliwice. Atari 130 XE. Komputery i software, technika video, żeglarstwo muzyka rozrywkowa.
Zbigniew Kowalczyk lat 35, ul. Gagarina 1/10, 93-530 Łódź. Video Genie System EG 3003. Hardware i software.
Sebastian Koszański lat 14, ul. Jaskrowa 18/1, Łódź. Commodore 64. Informatyka matematyka, gry.
Krzysztof Wilk lat 30, ul. Iwaszkiewicza 6/ 27, 39-100 Ropczyce (woj. Rzeszów) Commodore C-16. Elektronika, informatyka.
Jan Lis lat 26, ul. Dźbowska 12, 42-200 Częstochowa. TI-99/4A. Układy elektroniczne (cyfrowe), technika mikrokomputerowa.
Mariusz Gorczyca lat 28, ul. Mielczarskiego 32, 97-400 Bełchatów. ZX Spectrum +. Sport samochodowy, komputery, książki z serii „Sensacje XX wieku".
Andrzej Dudzicz lat 29, ul. Dzierżyńskiego 20A/4, 78-200 Białogard (woj. koszalińskie). Commodore 116, Meritum II, magnetofon 1531. Elektronika, muzyka elektroniczna.
Andrzej Wojda lat 18, ul. Cyranowska 11 A, 39-300 Mielec. Atari 800 XL. Informatyka, elektronika.
Marcin Karasiński lat 17, ul. Wyzwolenia 16/1, 41-907 Bytom. Colour Genie EG 2000, stacja dyskietek 51/4. Informatyka, elektronika, żeglarstwo, narciarstwo.
Bartosz Walaszek lat 13, ul. Wita Stwosza 16/1, Gdańsk-Oliwa. ZX-81. Komputery, astronomia.
Zenon Jackowiak lat 56, ul. H. Pobożnego 9/7, 70-508 Szczecin. Commodore VIC-20. Krótkofalarstwo.
Mariusz Kaleta lat 22, ul. Czarnowiejska 75/17, 30-049 Kraków, Colour Genie EG 2000 16 K RAM. Elektronika, elektroakustyka, informatyka.
Tomasz Ziegler lat 13, ul. Sienkiewicza 29/5, 45-037 Opole. Schneider 464. Oprogramowanie i gry komputerowe.
Mirosław Stefański lat 38, ul. Krakowska 13/3, 63-510 Mikstat (woj. kaliskie). Commodore 64. Hardware (rozbudowa posiadanego sprzętu), software (zastosowania profesjonalne).
Sebastian Caputa lat 13, ul. Rydygiera 20e/72, Toruń. Commodore 64. Komputery, informatyka, języki obce.
Ewa Pawłowicz, ul. Dekerta 3/3, 80-262 Gdańsk. CGL-M5 HOME COMP. oraz ZX 81 (16K). Informatyka, turystyka, muzyka, kino, teatr, literatura, mikrokomputery, języki programowania.
Tomasz Szefner lat 14, os. Lecha 16/55, 61 -293 Poznań. ZX 81 (Timex Sinclair, 2K). Informatyka, sport, historia.

Drogi Bajtku!



Na listy czytelników odpowiada Marcin Waligórski

Czy pomożecie mi w poznaniu mojego ZX-81 + 16 k? w jaki sposób zrealizować data i read? co oznaczają, i jak można użyć FAST i SLOW jak zapisać na zx-81 poniższy fragment programu:

```
10 REM Wyścigi psow
20 REM =========
30 DATA 4,6,132,252,124,202,169,169
40 REM Definicja znaku graficznego
50 FOR n = 0 TO 7
60 READ bajt
70 POKE USR „B" + n, bajt
80 NEXT n
```

Cezary Sadek
ul. Derdowskiego 22/21
71–087 Szczecin

BASIC komputera ZX-81 nie jest w ogóle wyposażony w instrukcje **READ** i **RESTORE** oraz w możliwość tworzenia zbiorów danych przy pomocy **DATA**. Z reguły struktury te używane są w programie w celu nadania wartości początkowych pewnym zmiennym, bądź też np. umieszczenia w pamięci komputera pewnych ciągów danych (jak w podanym powyżej przykładzie). Najprostszym i często niezadowalającym sposobem zastąpienia struktury używającej instrukcji **DATA** i **READ** jest w powyższych przypadkach dokonanie przypisań przy użyciu ciągu instrukcji podstawienia (**LET**) oraz umieszczanie ciągów bajtów w pamięci poprzez bezpośrednie użycie ciągu instrukcji **POKE**. Na przykład podany fragment programu możemy zapisać jako:

```
10 REM a0 — adres początkowy ciągu
20 LET a0 = ....
30 POKE a0,4 POKE a0 t 1,6
40 POKE a0 + 2,132: POKE a0 + 3,252
50 POKE a0 + 4,124: POKE a0 + 5,202
60 POKE a0 + 6,169: POKE a0 + 7,169
```

Widzimy, że rozwiązanie takie jest mniej wygodne w użyciu i zwiększa objętość programu.

BASIC komputera ZX-81 nie przewiduje możliwości definiowania znaków graficznych spoza ich standardowego zbioru. Z tego powodu nie jest dostępny na tym komputerze tryb definiowania własnych znaków podobny jak w Spectrum. Przełożenie przytoczonego w liście programiku tak, aby uzyskać analogiczny efekt, wymaga więc zastosowania bardziej złożonych metod.

ZX-81 posiada możliwość programowego wyboru trybu pracy. Tryb **FAST** pozwala na przyspieszenie wykonywania obliczeń poprzez wygaszenie ekranu. Użycie **SLOW** przywraca sytuację standardową.

Jestem zainteresowany możliwościami wykorzystania ZX Spectrum 48 w łączności krótkofalarskich. zainteresowany jestem m.in. programami przeistaczającymi spectrum w klawiaturę nadawczą morse'a...

Dariusz Toruń
ul. Związku Walki Młodych 6/41
02-786 Warszawa

W tym przypadku sprawa jest prosta ZX Spectrum, pomimo wyposażenia w prymitywny — niestety — generator akustyczny daje się z powodzeniem zastosować do wymienionego celu. Co więcej, napisanie programu w BASIC-u przekładającego ciąg znaków alfanumerycznych na ciąg dźwięków w alfabecie Morse'a nie jest zadaniem trudnym nawet dla początkującego programisty. Nie uzyskamy jednak w ten sposób dużej szybkości transmisji tekstu — spowodowane to jest powolnym działaniem interpretera języka BASIC. Gdyby uzyskana prędkość okazała się w praktyce niewystarczająca, należy do napisania takiego programu użyć jednego z języków kompilowanych (na Spectrum mamy np. Pascal) lub kodu maszynowego.

Sądzę, że program tego typu wraz z możliwością podłączenia wzmacniacza częstotliwości akustycznych stanowić może pomoc w pracy krótkofalarza.

Znacznie trudniejszym zadaniem byłoby napisanie programu umożliwiającego odbiór sygnałów alfabetu Morse'a i przedstawienie go na ekranie monitora w postaci tekstu. Do rozważenia tego problemu potrzeba kilku dodatkowych założeń, np. o częstotliwości odbieranego dźwięku.

Czytałem gdzieś o możliwościach uzyskania 22 kolorów na Spectrum. Niestety, nie doczytałem się, w jaki sposób można tego dokonać (przeglądałem jedynie pożyczoną książkę). Sam jestem widocznie zbyt słabym jeszcze programistą. Jak można otrzymać te kolory?

Jacek Kieniewicz
(adres do wiadomości redakcji)

Owszem, możliwość taka istnieje, tyle że ... polega nie na faktycznej możliwości uzyskania więcej niż 8 kolorów, lecz na nieskomplikowanym tricku. Możliwość ta jest też opisana w instrukcji użytkownika Spectrum. Definiujemy mianowicie pewien znak graficzny (np. GRAPHICS A) jako siatkę punktów ciemnych i jasnych. Kolor „tła" i „druku" możemy dobierać dowolnie (w zakresie 0..7 !) dla każdego pola znakowego. Np. wybór pary 0 (czarny) i 2 (czerwony) spowoduje, że wydrukowany na ekranie nasz znak sprawi wrażenie prostokąta o kolorze brązowym.

Oczywiście użycie tego rozwiązania nie pozwala na użycie wysokiej rozdzielczości, czy nawet pisanie tekstu w owych 22 kolorach.

Marcin

W numerze 5-6 z br. został zamieszczony fantastyczny program na ZX Dpectrum pt. „Numeryczna metoda znajdowania wartości pierwiastków funkcji". W trakcie wpisywania poszczególnych wierszy nie udaje mi się wpisać następujących pozycji.

370 LET xmax=MEM(x max,x(u))
380 LET ymax=MEM(y max,y(u))

Zbigniew Czechowski ul. Engelsa 8/1 59-220 Legnica

Niestety podczas listowania programu na drukarce nastąpił nie zauważony przeze mnie błąd, za co serdecznie przepraszam. Właściwa postać wierszy 370 i 380 jest następująca:

370 LET xmax = FN M(xmax , x(n)) 380 LET ymax = FN M(ymax , y (n))

Tomasz Kostrzyński

KLAN NIETYPOWYCH

Jestem posiadaczem mało znanego w Polsce mikrokomputera THOMSON MO5E.

Na razie nie posiadam żadnego oprogramowania firmowego ani literatury, a programy pisane w BASICU-u nie zadowalają mnie ze względu na ich powolność. Chciałbym sią dowiedzieć, czy jest może dostępna w Polsce literatura i oprogramowanie na ten mikrokomputer.

Pragnąłbym też zdobyć informacje o mikroprocesorze 63809 E, który jest jednostką centralną w tym mikrokomputerze.

Andrzej Foik
ul. Przodowników Pracy 90/2
47-223 Kędzierzyn-Koźle 5

GIEŁDA (ceny na dzień 1986-10-05)

		GIEŁDA BAJTKA (tys. zl)	KOMIS (tys. zl)	AUSTRIA (średnie) (öS)	FRANCJA (średnie) (FF)	RFN (średnie) (DM)	WLK. BRYT (średnie) (£)
SINCLAIR	ZX81	25–35	–	600	–	75–90	–
	ZX Spectrum 48 kB	70–85	95	1600	–	150–250	45–65
	ZX Spectrum Plus	100–115	140–160	1790	1350	200–300	70–90
	ZX Spectrum 128 kB	190	–	–	2200	500–530	150
	Drukarka SEIKOSHA GP50S	70–90	110	–	–	260	60–65
	Interface Kempston	5–13	20	250	300	35	7–9
	Stacja dyskietek OPUS 1 (3,5”)	150	–	500	–	395	95–105
COMMODORE	C-64	140–150	160–160	3600	1900	370–510	90–110
	C-128	265–310	430	6900	2890	650–700	210–230
	C-128D	600	–	13900	6850	1250	390–410
	Magnetofon 1531	30–35	40	900	350	80–90	–
	Stacja dyskietek 1541	155–170	200	4690	1950	470	110–150
	Stacja dyskietek 1570	–	300	6900	2300	490–540	160–170
	Drukarka MPS 801	120–130	150	2900	2200	199	–
	Drukarka MPS 803	160	200	–	2800	300–330	140–150
	Dyskietki 51/4 (średnia jak.)	0.7–1.5	1–2	16–25	7	0.9–0.4	0.8–2
ATARI	800 XL	65–80	90–100	1500	900	140–180	60
	130 XE	130–140	190	2100	1400	360	150
	Stacja dyskietek 1050	130–140	160–200	2200	2150	395	130
	Magnetofon	25–30	30–35	800	–	75–85	20
	Drukarka 1029	160	200	1990	–	–	85
AMSTRAD	464 z mon. monochromat.	195-210	300	6500	2690	598–720	170–190
	464 z monitorem kolor.	290	350–400	10000	3990	1100-1200	260–268
	6128 z mon. monochromat.	350–390	700–800	13000	3990	1450-1500	250–260
	6128 z mon. kolor.	485	1 mln	17000	5290	1900–2000	340–350
	PCW 8256	–	1.3 mln	–	5920	1795	380–400
	Dyskietki 3”	3–5	6	–	35	10–15	3.5–4
	Stacja dyskietek 3" do 464	210	295	6000	260	700	140

Cześć Maluchy!

Znamy się dość długo i myślę że już najwyższy czas spróbować jak dacie sobie radę z nieco trudniejszym zadaniem. Wszystkie dotychczasowe programy pisane były tak, by bez większych zmian mogły działać na każdym mikrokomputerze. Tym razem jest inaczej. Żeby uruchomić wasz program będziecie musieli nieźle ruszyć głową.

Mogłoby się wydawać, że w dobie łączności satelitarnej i komputerów, wielki wynalazek pana Morse-'a, jego słynny alfabet, powinien trafić do lamusa. Okazuje się jednak, że nie nastąpi to tak prędko. Zasadę zna każde dziecko — grupy krótkich i długich sygnałów dźwiękowych zastępują litery. Kłopot tylko w tym, żeby nauczyć się rozumieć ten ciąg pisków.

Nasz program służy właśnie do tego. Uczeń może wybrać sobie jedną z pięciu prędkości nadawania. Może również skorzystać ze ściągawki w postaci graficznego przedstawienia sygnałów (kropki i kreski) lub z niej zrezygnować. Komputer nadaje kolejno pięć losowo wybranych słów. Po każdym słowie zaczyna odliczać czas. Teraz odpytywany musi możliwie najszybciej napisać odczytane słowo i na koniec nacisnąć klucz spacji. Czasy poszczególnych odpowiedzi są sumowane, oczywiście za każdym razem komputer sprawdza prawidłowość odpowiedzi i wysyła na ekran odpowiedni komunikat. Za każdy błąd dolicza umowną ilość sekund. Po pięciu próbach następuje obliczenie punktów karnych i wydruk końcowego wyniku na ekranie. Egzamin skończony.

Popatrzmy jak działa nasz program. W linii 100 deklarujemy tablicę alfabet$ (37,1), w której umieścimy alfabety: łaciński i Morse'a. I tu pierwsza uwaga dla użytkowników Spectrum. Waszą tablicę musicie oznaczyć jedną literą (nie licząc znaku $). Wszystkie zmienne tekstowe — oznaczone zna kiem $ — występujące w programie musicie również zamienić na jednoliterowe. Uważajcie przy tym, by różnych zmiennych nie oznaczyć tą samą literą. Wczytywanie liter alfabetów odbywa się w pętlach (110-150).

```
99 REM **** zapamietanie
alfabetow ****
100 DIM alfabet$(37,1)
110 FOR i=0 TO 1
120   FOR j=1 TO 37
130 READ alfabet$(j,i)
140    NEXT j
150 NEXT i
499 REM **** obliczenie liczby
slow ****
500 LET liczbaslow=0
510 RESTORE 6030
520 READ slowo$
530 IF slowo$="koniec" THEN GOTO
1000
540 LET liczbaslow=liczbaslow+1
550 GOTO 520
999 REM *** wybor stopnia
trudnoci ***
1000 CLS
1010 LET dlugpisku=0
1020 PRINT
1030 INPUT "stopien trudnosci
(1/2/3/4/5 )";stopien
1040 IF stopien=1 THEN LET
dlugpisku=15
1050 IF stopien=2 THEN LET
dlugpisku=12
1060 IF stopien=3 THEN LET
dlugpisku=9
1070 IF stopien=4 THEN LET
dlugoisku=6
1080 IF stopien=5 THEN LET
dlugpisku=3
1090 IF dlugpisku=0 THEN GOTO
1020
1100 LET ekran=0
1110 INPUT "znaki na ekranie
(t/n)";odp$
```

TYLKO DLA PRZEDSZKOLAKÓW

KUBUŚ TELEGRAFISTA

W następnej kolejności komputer sprawdza iloma słowami do losowania dysponuje. Oczywiście moglibyśmy mu podać tę liczbę sami, lecz trzeba by wówczas pamiętać o jej zmianie po każdym uzupełnieniu danych. A tak wystarczy dopisać w dowolnej linii o numerze od 6030 do 6999 nowe słowa po instrukcji DATA i o resztę martwi się już sam komputer.

Możemy teraz przejść do właściwej części naszego programu czyli testu (1500-1960). Na początek losowany jest numer słowa (1550). Ponieważ instrukcje generujące liczby losowe mają w różnych mikrokomputerach różne konstrukcje, musicie zastosować właściwą dla waszego. Zmienna numerslowa powinna przyjmować wartości od 1 do wcześniej obliczonej liczby słów (zmienna liczbaslow).

Wylosowane słowo należy teraz odczytać i podstawić pod zmienną slowo$. W tym celu za pomocą instrukcji RESTORE 6030 w linii 1580 ustawiamy początek odczytywania (READ) od tej właśnie linii. W pętli od 1 do numeru słowa podstawiamy kolejno wszystkie dane pod zmienną slowo$. Jej ostatnia wartość będzie wylosowanym słowem.

Każde słowo musimy podzielić na litery. Konstrukcja w linii 1640 powoduje podstawienie pod zmienną literaslowa$ j-tej litery słowa. Spectrum nie posiada niestety instrukcji MID, ma za to podobnie działającą instrukcję TO.

Pętla 1650-1730 to wyszukiwanie danej litery w alfabecie Morse'a (1660), wyświetlenie odpowiedniego znaku na ekranie (1670) jeśli oczywiście ta opcja została wcześniej wybrana, podział znaku Morse'a na kreski i kropki (1690) tak samo jak w przypadku słowa, oraz odesłanie do odpowiednich podprogramów dźwiękowych — kreska i kropka (1700 i 1710). Po zakończeniu tej pętli wywoływany jest podprogram przerwy dźwiękowej.

Po nadaniu i ewentualnie wyświetleniu na ekranie całego słowa komputer przechodzi w stan oczekiwania na odpowiedź (1810-1870) równocześnie mierząc jej czas (1820). Odczytywanie klawiatury odbywa się za pomocą instrukcji INKEY$ (1810). W Commodore należy zastosować GET.

Koniec odpowiedzi sygnalizujemy naciskając klawisz spacji. Teraz następuje ocena odpowiedzi (1900-1910). Jeśli został popełniony błąd, komputer za karę dolicza czas (1950).

Test kończy się po nadaniu pięciu słów. Na podstawie czasu odpowiedzi obliczana jest liczba karnych punktów (2010 i 2020). Na koniec drukowany jest komunikat i pytanie, czy gramy ponownie (2000-2070).

Warto jeszcze napisać kilka słów o podprogramach dźwiękowych. Podstawą określenia długości dźwięku jest wybrana na początku programu dlugpisku, czyli długość brzmienia kropki. Kreska jest trzykrotnie dłuższa. Przerwa pomiędzy poszczególnymi kreskami i kropkami powinna wynosić w literze tyle co kropka a pomiędzy literami tyle co kreska. Teraz więc możemy wyjaśnić działanie poszczególnych podprogramów:

KRESKA — pisk o długości 3*dlugpisku i cisza 1*dlugpisku.

KROPKA — pisk o długości 1*dlugpisku i cisza 1*dlugpisku.

PRZERWA — cisza o długości 3*dlugpisku.

Uwaga dla posiadaczy komputerów, w których dźwięk jest generowany niezależnie od wykonywanego programu (Commodore, Amstrad, Atari). Warto wprowadzić mechanizm, który będzie zatrzymywał program na okres wybrzmiewania dźwięku.

Niektórzy z was zapewne domyślili się, że prezentowana wersja tego programu została napisana na mikrokomputerze Amstrad. Tak więc tylko posiadacze tych komputerów mogą „wklepać" ten program bez żadnych zmian. Pozostali muszą trochę popracować ale będą mieli okazję nauczyć się kilku nowych rzeczy. Myślę, że się opłaci!

Romek

KONKURS

"MIKRUSEK"

Ogłaszamy wielki konkurs na najlepszy program napisany przez dzieci — "MIKRUSEK". Do udziału zapraszamy wszystkie dzieci do lat 12 — nagrodą jest NIESPODZIANKA!!!

A oto warunki konkursu:

Program powinien być ułożony na jeden z następujących komputerów: COMMODORE, ATARI, AMSTRAD, SPECTRUM. Wydruk programu powinien być wykonany w miarę możliwości na drukarce; jeżeli jest to niemożliwe prosimy o nadsyłanie wyraźnych, bardzo starannie przepisanych, najlepiej na kartce papieru w kratkę. "BAJTEK" nie stawia wymogów tematycznych dla "MIKRUSKA" — mogą to być gry, programy użytkowe, czy edukacyjne. Punktowane będą: poprawność i czytelność programu, zastosowanie i przydatność oraz zwięzłość (nie dotyczy komentarzy REM!) ale przede wszystkim pomysł. Najlepsze programy otrzymują nagrody oraz zostaną wydrukowane w "BAJTKU".

Termin nadsyłania programów konkursowych mija 1 LUTEGO 1987 ROKU.

```
1120 IF odp$="t" THEN LET ekran=1
1499 REM ************ test
************
1500 LET czas=0
1510  FOR i=1 TO 5
1520 FOR t=1 TO 1000
1530 NEXT t
1540 CLS
1550 LET numerslowa=INT(RND(1)*licz
baslow+1)
1560 RESTORE 6030
1570 PRINT
1580 PRINT
1590 FOR j=1 TO numerslowa
1600 READ slowo$
1610 NEXT j
1620 IF ekran=1 THEN PRINT "/";
1630 FOR j=1 TO LEN(slowo$)
1640 LET
literaslowa$=MID$(slowo$,j,1)
1650 FOR k=1 TO 37
1660 IF alfabet$(k,0)<>literaslowa$
THEN GOTO 1730
1670 IF ekran=1 THEN PRINT
alfabet$(k,1); " /";
1680 FOR l=1 TO LEN(alfabet$(k,1))
1690 LET
pisk$=MID$(alfabet$(k,1),1,1)
1700 IF pisk$="-" THEN GOSUB 5000
1710 IF pisk$="." THEN GOSUB 5100
1720 NEXT l
1730 NEXT k
1740 GOSUB 5200
1750 NEXT j
1760 LET odczyt$=""
1770 PRINT
1780 PRINT
1790 PRINT "???????????????????"
1800  PRINT
1810   LET literaodczyt$=INKEY$
1820   LET czas=czas+1
1830    IF literaodczyt$="" THEN
1810
1840    IF literaodczyt$=" " THEN
1880
1850   PRINT literaodczyt$;
1860    LET
odczyt$=odczyt$+literaodczyt$
1870    GOTO 1810
1880  PRINT
1890 PRINT
1900 IF odczyt$=slowo$ THEN PRINT
"brawo!"
1910  IF odczyt$<>slowo$ THEN GOTO
1930
1920  GOTO 1960
1930 PRINT "blad!"
1940 PRINT slowo$
1950 LET czas=czas+5000
1960 NEXT i
1999 REM ******* zakonczenie gry
*******
2000 PRINT
2010 LET pkt=INT(czas/100)-20
2020 IF pkt<0 THEN LET pkt=0
2030 PRINT pkt; " pkt. karnych."
2040 PRINT
2050 INPUT "jeszcze raz";odp$
2060 IF odp$<>"nie" THEN GOTO 1000
2070 END
4997 REM
4998 REM ********* podprogramy
*********
4999 REM *********** kreska
***********
5000 SOUND 1,50.3*dlugpisku,15
5010 SOUND 1,0,dlugpisku,0
5020 IF SQ(1)>=128 THEN GOTO 5020
5030 RETURN
5099 REM *********** kropka
***********
5100 SOUND 1,50,dlugpisku,15
5110 SOUND 1,0,dlugpisku,0
5120  IF SQ(1)>=128 THEN GOTO 5120
5130 RETURN
5199 REM *********** przerwa
***********
5200 SOUND 1,0,3*dlugpisku,0
5210 IF SQ(1)>=128 THEN GOTO 5210
5220 RETURN
5998 REM
5999 REM ************ dane
************
6000 DATA a,b,c,ch,d,e,f,g,h,i,
j,k,l,,m,n,o,p,r,s,t,u,v,w,x,y
,z,1,2,3,4,5,6,7,8, 9,0
6010 DATA . _ , _ . . . , _ . _ . ,
_ _ _ _ , _ . . , . , . . _ . , _ _
. , . . . . , . . , . _ _ _ , _ . _
, . _ . . , _ _ , _ . ,
_ _ _ , . _ _ . , _ _ . _ , . _ . ,
. . . , _ , . . _ , . . . _ , . _ _
, _ . . _
6020 DATA  _ . _ _ , _ _ . . , . _
_ _ _ , . . _ _ _ , . . . _ _ , . .
. . _ , . . . . . , _ . . . . , _ _
. . . , _ _ _ . . , _ _ _
 _ . , _ _ _ _ _
6030 DATA bajtek,kuba,komputer,kocz
kodan,list,litera,telegraf,rzeka,dr
zewo,krokodyl
7000 DATA koniec
```



ELEKTRONICZNA HOSTESSA

(KORESPONDENCJA Z EXPO-86 W VANCUVER)

SĄ niesłychanie praktyczne. Nie miewają humorów, nie potrzebują modnych strojów, są bardzo, bardzo atrakcyjne, a posilają się jedynie kanadyjskim prądem o napięciu 110V.

Takich elektronicznych hostess, które wspaniale opiekują się gośćmi wystawy światowej EXPO-86 w Vancouver. odpowiedzą niemalże na każde ich pytanie dotyczące pawilonów, atrakcji, cen czy wolnych miejsc hotelowych, światowy potentat komputerowy f- koncern IBM ustawił na 70 ekspozycyjnych hektarach aż 100. Całość jest na bieżąco programowana, bo przecież w dziennych schematach imprez na EXPO zachodzą zmiany, ale tego przeciętny widz, który nabył całodzienny bilet za 20 dolarów i teraz zatrzymał się przed zachęcająco mrugającym ekranem już nie zobaczy. Ma bowiem do czynienia z końcowym elementem systemu.

KOMPUTERY, KOMPUTERY I ZEGAREK

Do infokiosków IBM trudno się dopchać. Każdy chce się przecież przekonać, jak bardzo zmyślna jest ta wizytówka elektronicznego giganta, tym bardziej, iż zabawa jest zupełnie bezpłatna.

Ciekaw jestem, jak wyglądałyby klawiatury, do których przyzwyczailiśmy się w Polsce, pod koniec wystawy (EXPO-86 trwa 165 dni — od 2 maja do 13 października), po katuszach tylu milionów palców (szacuje się, że wystawę odwiedzi około 20 milionów ludzi). Zamiast tego wszystkiego dzieci i dorośli porozumiewają się z komputerową siecią informacyjną wystawy dotykając odpowiednio oznaczonych punktów na powierzchni kolorowego monitora.

W ten sposób wywołać można z pamięci maszyny krajobrazy 54 krajów uczestniczących w wystawie, 7 kanadyjskich prowincji i 2 terytoriów niezależnych oraz 3 amerykańskich stanów prezentujących się oddzielnie. To oczywiście tylko niezmiernie drobny ułamek potężnych możliwości systemu. Można bowiem zapytać, co działo się na EXPO na przykład 7 maja o 13.25 lub też, czego możemy oczekiwać 7 października o tej samej porze. Ba, są nawet uaktualniane cały czas spisy potraw podawanych w poszczególnych narodowych restauracjach wraz z cennikiem, czy też życiorysy szefów poszczególnych ekspozycji.

Trudno opowiedzieć o wszystkich elektronicznych cudeńkach, które zaparły dech waszemu autorowi, na pobieżne zwiedzanie EXPO-86 trzeba bowiem przeznaczyć co najmniej trzy dni.

25-metrowy papuziożółty zegarek naręczny oplata pudło pawilonu szwajcarskiego. W bransolecie okalającej dach umieszczono serce tej niezwykle przemyślanej maszynerii, która odmierza upływający czas z dokładnością nie spotykaną w zupełnie normalnych chronometrach. 25-metrowa elektroniczna Omega może mieć, co najwyżej, jednosekundową roczną tolerancję. „Wzorzec" cieknących sekund, minut i godzin przesyłany jest do Vancouver, nad brzeg Pacyfiku ... ze Szwajcarii poprzez ... satelitę. Kitka ogniw wielkiej bransolety, to nic innego jak precyzyjne anteny odbierające sygnał sputnika. O tym wszystkim można się przekonać dopiero wewnątrz budynku. Na zewnątrz nic nie świadczy o tym, że jaskrawy szwajcarski gigant ma cały czas kontakt z kosmosem.

ZABAWKI DLA DUŻYCH DZIECI...

Japończycy nie pokazują jeszcze komputerów piątej generacji, z którymi można byłoby przeprowadzić uczoną dysputę na tematy filozoficzne w dowolnym języku, lecz własne rozwiązania komunikacyjne w skali 1/100. Miniaturowe samochody, tramwaje, statki, pociągi zwykłe i superszybkie, a wreszcie samoloty podwieszone do dających im energię instalacji na czarnym jak smoła suficie — wszystko to porusza się w sposób harmonijny dzięki mikroprocesorom ukrytym gdzieś w czeluściach tego całego ciągu żyjących makiet. Ten gadget jest dla wszystkich. Specjaliści poproszą o szczegóły, a wówczas otrzymają wszystko na co Nippon stać, z możliwością rozmowy z komputerem wyładowanym do granic możliwości danymi włącznie. W osobnym pomieszczeniu, nazwijmy je bibliotecznym, pyszna skośnooka dziewczyna sadza ciekawego przy oddzielnym monitorze i wprowadza do sprzężonego z nim komputera odpowiedni dysk, czy też włączy kasetę video. Całego Kraju Wschodzącego Słońca do Vancouver przewieźć się nie udało, ale nie wiem, czy ktoś zdołałby w ciągu owych 165 dni zapoznać się ze wszystkim, co zostało zarejestrowane na dostępnych w pawilonie japońskim nośnikach informacji.

... I NIE TYKO ZABAWKI

Na ubiegłorocznym salonie lotniczym w Paryżu Francuzi pokazali swoje najnowsze odrzutowe dziecko — myśliwiec bombardujący „Rafale". Absolutną nowością było całkowite wyeliminowanie z kabiny pilota wszelkich, tak typowych do tej pory, zegarowych wskaźników. Zastąpiły je ekrany pokładowych komputerów. Pilot będzie ostrzegany przed niebezpieczeństwami... ludzkim głosem i sam otrzymał możliwość wydawania już tylko ustnie poleceń swojej maszynie. Republika Federalna Niemiec pokazała w Vancouver, że podobne urządzenie nie musi koniecznie służyć Marsowi.

Co kilkadziesiąt minut w zachodnioniemieckim pawilonie odbywały się pokazy prowadzenia Mercedesa 190E, wozu zupełnie bez kierownicy. Inwalida pozbawiony władz w górnych kończynach bez trudu poradzi sobie z tym samochodem. Nogi sterują układem kierowniczym, inne polecenia — wydane głosem — odbiera komputer, a następnie przekształca je w impulsy przekazywane różnym układom pojazdu.

Na koniec dwie ciekawostki. Sporym zaskoczeniem dla Amerykanów, Kanadyjczyków, a i również dla piszącego te słowa był chińsk profesjonalny komputer osobisty o barwnej nazwie „Wielki Mur", z własną drukarką, kolorowym monitorem i stacją dysków. Na oko, nie demonstrowano go w akcji, nie odbiegał niczym od modeli firm Apple, czy IBM. Niespor dzianką był również fakt, iż jugosłowiańska spółka ISKRA (od żarówek po mikroprocesory) eksportuje do ... Stanów Zjednoczonych swoje monitory komputerowe i inne elementy techniki mikroelektronicznej.

Wojciech Łuczak





www.t2e.pl

Seria „Reduks” ma na celu przetworzenie – pełne zdigitalizowanie prasy komputerowej z początków okresu informatyzacji naszego kraju.

W odróżnieniu od plików w formacie djvu i cbr gdzie format papierowy jest tylko zeskanowany i połączony w jeden dokument, **reduks polega na ponownym przełamaniu dokumentu, wraz z ponowną obróbką graficzną - często z ich ponowną kreacją**. Tam gdzie to możliwe zachowana jest grafika z pierwowzoru, jednak jeśli materiał jest zbyt złej jakości - zostaje on podmieniony (oczywiście jeśli jest dostępny) na równoważny. Jeżeli uda się odnaleźć błąd w zamieszczonych listingach są one poprawiane tak, aby mogły zostać uruchomione na emulatorach lub mikrokomputerach.

Rozpocząłem ten cykl od „Bajtków”, jednak mam nadzieję, że uda mi się wzbogacić kolekcję zdigitalizowanych i ponownie przełamanych periodyków także o inne ważne tytuły z lat ubiegłych, o których powoli zapominamy, a bez których współczesny obraz komputeryzacji wyglądałby zgoła inaczej.

Wydania „Reduksowe” są bezpłatne, nie roszczę sobie żadnych praw do zawartości przetwarzanych magazynów, jednak ilość czasu i pracy jaką wkładam aby mogły one trafić w Twoje ręce jest na tyle spora, że pozwolę sobie napomknąć iż możesz wesprzeć mnie przekazując dotację w dowolnej wysokości.

Jeśli posiadasz materiały fotograficzne lub edytorskie, które są lepszej jakości niż prezentowane w numerach, daj znać.



W ramach wolnego czasu odtwarzam stare magazyny komputerowe i prowadzę serwis emulacyjny try2emu. Jeżeli uważasz, że moja publiczna działalność możesz postawić mi kawę, a kawę lubię;)



https://buycoffee.to/t2e

Osoby, które wspierają serwis datkami* mają dostęp do reduksów w wysokiej rozdzielczości z usuniętymi reklamami.

W zależności od wysokości datku każdy otrzymuje czasowy dostęp do reduksów w pełnej rozdzielczości na miesiąc, 3 miesiące lub 6 miesięcy.
Osoby, które aktywnie wspierają projekt mają dostęp nieograniczony.